Developing the Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap

Final Technical Report SERC-2013-TR-016-3

December 31, 2013

Principal Investigator -Dr. Jon Wade, Stevens Institute of Technology

Co-Principal Investigator- Dr. William Watson, Purdue University

Dr. Douglas Bodner - Georgia Tech

Dr. George Kamberov - Stevens Institute of Technology

Dr. Richard Turner - Stevens Institute of Technology

Brent Cox - Stevens Institute of Technology

Arya Irani - Georgia Tech

With Contributions by Subject Matter Experts

Eric (Rick) Abell, John Griffin, and John McKeown

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **31 DEC 2013**	2. REPORT TYPE **Final**	3. DATES COVERED
4. TITLE AND SUBTITLE **Developing the Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap**		5a. CONTRACT NUMBER **H98230-08-D-0171** 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Wade /Dr. Jon**		5d. PROJECT NUMBER **RT 16-3** 5e. TASK NUMBER **TO 06** 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology, Purdue University, Georgia Institute of Technology**		8. PERFORMING ORGANIZATION REPORT NUMBER **SERC-2013-TR-16-3**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **DASD (SE), DAU**		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited.**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

This document is a summary of the work that was completed in the second increment year of the SERC Research Topic 16 "Developing Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap". The major research activities in Increment 2 are as follows: 1.Update Documentation and Planning 2.Prototype Evaluation (cont.) 3.Pilot System Development: 4.Pilot System Evaluation 5.Open Source Preparation and Deployment 6.External Developers Engagement 7.Develop Multi-Learner Technology 8.Write Final Report In addition to the work activities, the following five top program risks were identified and tracked throughout Increment 2 of the program: 1.Project Management - Inability to support known and evolving customer/user feedback with current staff, budget and timeframe. 2.Configuration Management - Inability to successfully manage the large number of files,configuration variables,present in the Experience Accelerator. 3.Technology Development - Inability to tradeoff long-term architecture and technology objectives (leading to successful open source support) vs. short-term prototype goals. 4.Content Development - Inability to produce a prototype that provides a compelling experience,supports the desired learning and is seen to be authentic. 5.Evaluation - Inconclusive results due to threats to validity of Experimental design(inability to generalize results),limited availability of suitable subjects and insufficient literature to support development of evaluation instruments. A set of lessons learned was compiled and categorized as noted below: œCompetencies, Learning and Content œComplexity/Effort vs. Authenticity/Learning œTechnology œR&D Processes

15. SUBJECT TERMS

<table>
<tr><td colspan="3">16. SECURITY CLASSIFICATION OF:</td><td rowspan="2">17. LIMITATION OF ABSTRACT
UU</td><td rowspan="2">18. NUMBER OF PAGES
70</td><td rowspan="2">19a. NAME OF RESPONSIBLE PERSON</td></tr>
<tr><td>a. REPORT
unclassified</td><td>b. ABSTRACT
unclassified</td><td>c. THIS PAGE
unclassified</td></tr>
</table>

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

Executive Summary

This document is a summary of the work that was completed in the second increment year of the SERC Research Topic DO1/TTO2/0016 "Developing Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap" supported by the Defense Acquisition University (DAU). The purpose of the research project is to test the feasibility of a simulated approach for accelerating systems engineering competency development in the learner. The SEEA research project hypothesis is:

> *By using technology we can create a simulation that will put the learner in an experiential, emotional state and effectively compress time and greatly accelerate the learning of a systems engineer faster than would occur naturally on the job.*

The major research activities in Increment 2 are as follows:

1. Update Documentation and Planning
2. Prototype Evaluation (cont.)
3. Pilot System Development:
   3.1. Refinement based on evaluation findings
   3.2. Architecture, Design, Technology and Tools for Flexibility & eventual Open Source
4. Pilot System Evaluation:
   4.1. Plan Update
   4.2. Learner Identification
   4.3. Prototype Evaluation
5. Open Source Preparation and Deployment:
   5.1. Prototype Completion
   5.2. Migration, Open Source Hosting & Development, and Ticketing
   5.3. Tool Completion
   5.4. Design Flow
   5.5. Documentation
6. External Developers Engagement
7. Develop Multi-Learner Technology
8. Write Final Report

In addition to the work activities, the following five top program risks were identified and tracked throughout Increment 2 of the program:

1. **Project Management** - Inability to support known and evolving customer/user feedback with current staff, budget and timeframe.
2. **Configuration Management** – Inability to successfully manage the large number of files, configuration variables, present in the Experience Accelerator.

3. **Technology Development** - Inability to tradeoff long-term architecture and technology objectives (leading to successful open source support) vs. short-term prototype goals.
4. **Content Development** - Inability to produce a prototype that provides a compelling experience, supports the desired learning and is seen to be authentic.
5. **Evaluation** - Inconclusive results due to threats to validity of Experimental design (inability to generalize results), limited availability of suitable subjects and insufficient literature to support development of evaluation instruments.

A set of lessons learned was compiled and categorized as noted below:

1. Competencies, Learning and Content
2. Complexity/Effort vs. Authenticity/Learning
3. Technology
4. R&D Processes

A DAU student pilot review was held on October 29-30, 2013 with the following sponsor representatives: James Anthony, Tony Costanza, Darren Dusza, Steven Jones, Scott Lucero, Dave Pearson, and John Snoderly. Despite a number of technical issues relating to networking capabilities and application stability and shortage of class time, the potential of the Experience Accelerator was validated through feedback from the students.

A number of the targeted lessons were clearly represented in the team presentations, an indication of the effectiveness of the EA in promoting its targeted learning outcomes. For example, for problem solving and recovery, the importance of *small* schedule delays and the use of additional staff to remediate the schedule problem was touched on by several teams. Teams 1, 3, and 5 mentioned the need to hire staff early in their lessons learned, while Team 2 called for more dramatic staff shifts. Team 4 highlights the results of slipping the schedule too much, lamenting that they were fired for slipping CDR.

Another example can be seen with “Cutting corners to make short term goals while ignoring long term outcomes” which stresses the need to make decisions early. This was touched on repeatedly by the teams in their lessons learned. Team 1 mentions hiring more staff early, Team 3 calls for shifting staff earlier, Team 4 notes how their early emphasis on software worked, and Team 5 reflects on the need to ramp up staff more quickly.

These examples demonstrate that for these two targeted learning outcomes in particular, nearly all of the teams learned the outcomes as they very clearly highlighted them in their presentations as the lessons they had learned. Other learning objectives were also highlighted by the different teams. These lessons were learned despite the fact that the learners only fully completed the first two phases of the experience before speeding through the remaining phases in order to see their results. Furthermore, the EA was designed to be played multiple times by learners, so these results are indicative of impressive learning gains given the limited implementation of the experience.

Students also were asked to provide their perceptions of the EA – what it did well and what could be improved. The class discussed these and the comments were captured. Comments highlighted a number of key features of the EA as positives. These included the case-based format of the EA, noting its representation of real-life issues and modeling of real work interactions. Students also noted the importance of immediate feedback on the decisions and the interactive nature of the simulation – accelerating learning by simulating a project lifecycle in a short amount of time. The challenging aspect of the simulation was also highlighted as it was noted that being fired kept the learning challenging a more interesting, a possible reference to the EA's targeted "scar tissue" – emotional connection in order to promote learner transfer of learned objectives. One key positive from the feedback was that the user interface received a great deal of praise – an important change from the SME feedback.

Recommendations provided additional insights on how we can further improve the EA. For example a greater focus on performance and technical aspects as opposed to cost and schedule was highlighted. A few small bugs were also identified which will be corrected, and information which will be included in future iterations of the instructors' manual will help to further improve the efficacy of the EA.

Ultimately, the formal evaluation of the EA was hindered by the inability to compare novice learner actions and thinking to that of expert SMEs. However, while this comparison could not be made in this implementation, it certainly could be done in a future implementation. Furthermore, clear evidence of learning was gathered from the data that was gathered and learner perspectives on the efficacy of the EA were largely positive while still providing some helpful suggestions for improvement. The formal evaluation can therefore be seen as a success and indicative of the EA's efficacy in meeting its targeted learning objectives.

Follow-on work has been defined for Increment 3 that is focused on the following:

- EA System Capabilities
    - Completion and stabilization of multi-learner mode
    - Provide means of informing learner of impact of recommendations
    - Ensure that dialog is synchronized with recommendations
    - Improve learner interface with status charts to eliminate need to page through entire set
- Tools
    - Create set of tools that allow the DAU to customize and create new Experiences
- Deployment Deliverables
    - Define explicit EA deliverables to support DAU deployment
- Hosting Requirements
    - Specify technical details of hosting requirements

Table of Contents

List of Figures

Preface

This document is a summary of the work that was completed in the Increment 2 of the SERC Research Topic DO1/TTO2/0016 "Developing Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap" supported by the Defense Acquisition University (DAU). This summary focuses on each of the work items noted in the proposal.

The following are the documents that were produced by this research and may be referenced in this document:

Experience Accelerator RT16 Project documents:

- RT16 Project Goals and Success Metrics (A013)
- RT16 Technical and Management Work Plan (A009)
- RT16 Monthly Status Reports (A008)
- Experience Accelerator Concept of Operations (A013)
- Experience Accelerator System Architecture and Design Specification (A013)
- Experience Accelerator Systems Specification (A013)
- Experience Accelerator: Experience Design Document
- Experience Accelerator White Paper
- Developing Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap, Increment 1 proposal
- Developing Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap, Increment 2 proposal
- Developing Logistics Experience Accelerator (EA) Prototype, 5/15/2012.
- Technical Leadership Development Experience Accelerator Prototype, 8/22/2012.
- Developing the Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap, Final Technical Report Year 1, SERC-2011-TR-19, May 31, 2011.
- Developing the Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap, Final Technical Report Year 1, SERC-2012-TR-xx, October 24, 2012.

Publications:

- Bodner, D., Wade, J. (2013) "Multi-Criteria Simulation of Program Outcomes," Proceedings of the 2013 IEEE International Systems Conference, 215-221, Orlando, FL, April 15-18, 2013.
- Bodner, D., Wade, J., Watson, W., Kamberov, G (2013) "Designing an Experiential Learning Environment for Logistics and Systems Engineering," Proceedings of the 2013 Conference on Systems Engineering Research (CSER), Atlanta, GA, March 20-22, 2013.
- Wade, J., Kamberov, G., Bodner, D., Squires, A. (2012) "The Architecture of the Systems Engineering Experience Accelerator", International Council on Systems Engineering (INCOSE) 2012 International Symposium/European Conference on Systems Engineering (EUSEC), Rome, Italy, July 9-12.

- Bodner, D., Wade, J., Squires, A., Reilly, R., Dominick, P., Kamberov, G., Watson, W. (2012), “Simulation-Based Decision Support for Systems Engineering Experience Acceleration”, IEEE Systems Conference, Vancouver, BC, Canada, March 19-23.
- Squires, A., Wade, J., Watson, W., Bodner, D., Reilly, R., Dominick, P. (2012), “Year One of the Systems Engineering Experience Accelerator”, Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER), Rolla, Missouri, March 19-22, 2012.
- Squires, A., Wade, J., Watson, B., Bodner, D., Okutsu, M., Ingold, D., Reilly, R., Dominick, P., Gelosh, D. (2011), “Investigating an Innovative Approach for Developing Systems Engineering Curriculum: The Systems Engineering Experience Accelerator”, Proceedings of the 2011 American Society for Engineering Education (ASEE) Annual Conference and Exposition, Vancouver, BC, Canada, June 26-29, 2011.
- Squires, A., Wade, J., Dominick, P., Gelosh, D. (2011) “Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers”, Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER), Redondo Beach, CA, April 14-16, 2011.
- Squires, A., Wade, J., Bodner, D., (2011), “Systems Engineering Experience Accelerator Workshop”, National Defense Industrial Association (NDIA) 2011 14th Annual Systems Engineering Conference, San Diego, CA, October 24-27. (Presentation)

1 INTRODUCTION

Systems engineering educators are struggling to address workforce development needs required to meet the emerging challenges posed by increasing systems complexity (Bagg, et. al, 2003) and the widening gap in systems engineering expertise in the workforce (Charette, 2008). The Systems Engineering Experience Accelerator (SEEA) research project was conceived as a critical response to these needs and challenges. The project was initiated to validate the use of technology to potentially create an experiential, emotional state in the learner coupled with reflective learning so that time is effectively compressed and the learning process of a systems engineer (SE) is significantly accelerated as compared to the rate at which learning would occur naturally on the job. The purpose of the research project is to test the feasibility of a simulated approach for accelerating systems engineering competency development in the learner. An example of how the various concepts developed for the SEEA are related is shown in Figure 1.

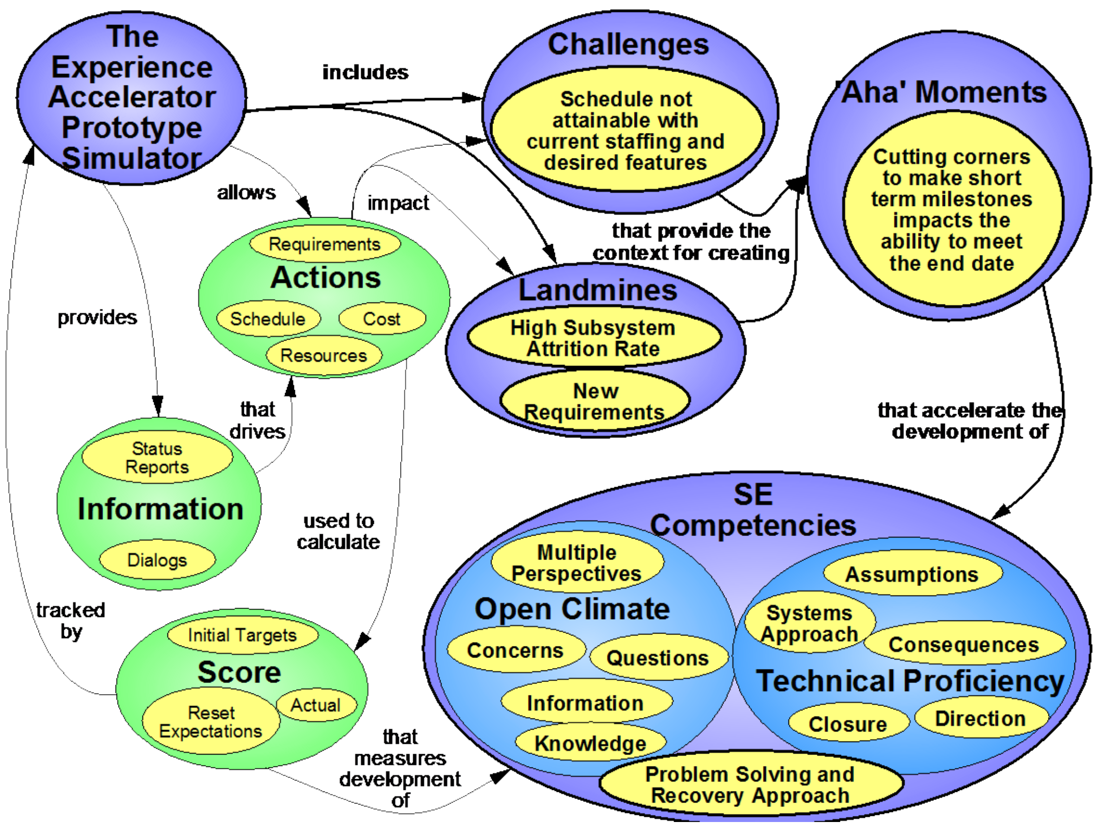


Figure 1: Notional Diagram of the SEEA Prototype Simulator

As shown, the development team had a threefold challenge to balance the development of the simulator technology that supports displayed content (shown in green) that, in turn, supports the developed concepts (shown in purple). The goal was to effectively create challenges and landmines that support the learner's experience of the necessary "Aha" moment. The intent was that by experiencing the "Aha" moment, the learner transitions to a more advanced level of understanding in the targeted competency, in this case "Problem Solving and Recovery Approach".

The testing of the prototype will support evaluation of the theoretical capabilities of the developed system and provide guidance for the continuing development of the SEEA simulator going forward.

1.1 Background and Motivation

In *The Art and Science of Systems Engineering*, Mr. Harold Bell, Director Advanced Planning and Analysis Division, NASA Office of Chief Engineer, is quoted as saying: "A great systems engineer completely understands and applies the art of leadership and has the experience and scar tissue from trying to earn the badge of leader from his or her team" (Ryschkewitsch, et. al, 2009, p. 2). Historically, competent systems engineers have developed their "scar tissue" by gaining the necessary insights and wisdom through both failures and successes, in an integrated real world environment. In the workplace, however, the learning events that result in the development of "scar tissue" are distributed, sometimes sparsely, over time. In addition, a common benchmark time for the development of a competent SE is a minimum of about 10-15 years (Dubey, 2006). Given there is a shortfall of SEs in the global workforce today (NDIA SE Division, 2010) and no readily available source of SEs to replace the top SEs in the retiring baby boomer generation, the time to develop competent SEs needs to be significantly shortened.

The primary goal of the SEEA, once it is developed, is to accelerate the maturation of SEs in the workforce by providing the opportunities to earn "scar tissue" through realistic, engaging simulation. These tailored experiences will allow the learner to feel the consequences of success and failure in a simulated environment so they can gain the necessary insights and wisdoms to mature as a SE, and yet not jeopardize the lives of others or compromise their careers. The initial target audience of the SEEA program is lead program SEs in the acquisition workforce who are required to effectively manage complex systems throughout their lifecycle from an acquisition/acquirer viewpoint in a typical program office. The initial focus is on maturing these leads to prepare them for executive assignments.

1.2 Project Goals and Success Metrics

Based on SEEA research team meetings and feedback provided by the sponsors, the team set specific goals and success metrics as summarized in the following sections.

1.2.1 Purpose

The ultimate purpose of the SEEA is to leverage technology to create an experiential, emotional state in the learner so that time is effectively compressed and the learning process of a systems engineer accelerated as compared to the rate at which learning would occur naturally on the job.

The purpose of this project is to develop a prototype of the SEEA that is focused on a small set of competencies, in order to evaluate the theoretical capabilities of that technology.

1.2.2 Hypothesis

By using technology we can create a simulation that will put the learner in an experiential, emotional state and effectively compress time and greatly accelerate the learning of a systems engineer faster than would occur naturally on the job.

1.2.3 Program Goals

The primary goal of the SEEA is to transform the development of systems engineers by creating a new paradigm capable of significantly reducing the time to mature and sustain a senior systems engineer while providing the skills necessary to address emerging systems challenges in an economically attractive manner.

The approach is to build insights and "wisdom" and hone decision making skills by:

- Creating a "safe", but realistic environment for decision making where decisions have programmatic and technical consequences
- Exposing the participants to job-relevant scenarios and problems
- Providing rapid feedback by accelerating time and experiencing the downstream consequences of the decisions made

Outcomes needed to achieve this goal include:

1. Moving the systems engineer to the next level of proficiency in one or more SE competencies as listed in the Systems Planning, Research Development, and Engineering (SPRDE) Systems Engineering (SE) and Lead Systems Engineer (LSE) competency model, known as the SPRDE-SE/LSE.
2. Developing and maturing systems thinking skills.
3. Developing and maturing leadership skills.

1.2.4 Target Audience

The initial focus is on the Systems Engineering Executive Level skills of a DoD Lead Systems Engineer necessary to effectively manage complex systems throughout their lifecycle from an acquisition/acquirer viewpoint in a typical Project Management Office (PMO). The skills addressed may well complement or support those taught in senior program management courses. The SEEA targets the entire life-long learning of the Systems Engineer.

1.2.5 SUCCESS METRICS

Success of the Experience Accelerator prototype will be indicated with a positive result in the following areas:

- Experienced Lead Program Systems Engineers authenticate the EA and provide useful feedback on areas of improvement.
- Learners express general satisfaction with the learning experience.
- The potential for learners who successfully complete the training to be able to immediately implement lessons learned from the training experience to the job, assuming the culture allows this.

1.3 MANAGEMENT PLAN/TECHNICAL OVERVIEW

The RT16 work plan is summarized in the next section, with the detail in the *Technical and Management Work Plan (A009)*. Program risks, addressed in the following section, were also reported on in the latter half of the project in the Monthly Status Reports (A008).

1.3.1 RESEARCH ACTIVITIES

The major research activities that were completed in Increment 2 are as follows:

1. **Increment 2 Planning:** This involved updating the Program objectives, feature set, capabilities and schedule, and Management Plan.

2. **Prototype Evaluation (cont.):** The preliminary assessment initiated in Increment 1: Phase 3 will be continued in Increment 2. This will involves testing the prototype with a broader set of expert users. This feedback will be used to continue to determine the deficiencies and strengths of the current system. These results will be used more finely tune the necessary improvements to the system for a more thorough evaluation with representative users. This activity will be completed at Purdue University and Stevens Institute of Technology **(Dr. Bill Watson, Dr. Richard Turner, Dr. Jon Wade)**

3. **Pilot System Development**: This work entails the further refinement of the prototype system based on feedback such that it can be used as a pilot for broader usage and evaluation. This development is targeted to result in a system that could be piloted in a select number of sponsor's classes as well as potentially classes at the researchers' institutions. Pilot learners will utilize the prototype in a variety of targeted contexts, including as part of formal coursework, as well as individually in informal settings. Development will also focus on facilitation and implementation manuals for a variety of contexts, including formal coursework and informal settings, in order to reduce adoption barriers and promote effective implementation of the system for maximized learning impacts. This work will be undertaken at Stevens, Georgia Tech and Purdue. **(Dr. Doug Bodner, Dr. George Kamberov, Dr. Richard Turner, Dr. Jon Wade)**

3.1. **Refinement based on evaluation findings.** This activity will be completed at Stevens, Georgia Tech and Purdue University. Activities will be directed by the results of the Prototype Evaluation and will focus on refining the Pilot.

3.2. **Architecture, Design, Technology and Tools for Flexibility & eventual Open Source –** This activity will be completed at Stevens Institute of Technology and will focus on refining an underlying architecture, design, technology and tools for the EA which will support flexibility and an eventual transition to an open source architecture.

4. **Pilot System Evaluation:** The assessment will be elevated to involve formal assessment of the impact of the prototype. This involves piloting the prototype with a representative sample of learners (students and practitioners). This needs to be done such that the results are indicative of what would likely be achieved with a finished delivery vehicle on a representative group of participants. Likewise, assuming available subject populations, comparative assessments of a variety of implementation approaches, such as instructor-led versus individual or instructor scaffolded versus in-simulation scaffolding will be conducted in order to guide best practices for implementation and facilitation. This activity will be completed at Purdue University and Stevens Institute of Technology **(Dr. Bill Watson, Dr. Richard Turner, Dr. Jon Wade)**

4.1. **Plan Update:** Metrics, associated instruments and evaluation experiments will be refined based on review feedback from the prototype. The result will be an update in the Plan for Formal Evaluation report.

4.2. **Learner Identification:** Determination of targeted learners at DAU and SERC collaborating universities will be identified. Dates and logistics will be determined to support the evaluation efforts.

4.3. **Prototype Evaluation:** This is the actual evaluation of the learners, as determined in the Evaluation Plan. These results will then need to be analyzed to determine the efficacy of the approach and how it will be refined for deployment.

5. **Open Source Preparation and Deployment:** This involving completing the prototype development such that it can easily be supported, migrating the prototype to an open source development location, implementing a ticketing system for bug tracking, completing a set of tools and example design flow, and providing a document set which can be updated by an open source community. This work will be undertaken at Stevens and Georgia Tech. **(Dr. Doug Bodner, Dr. George Kamberov, Dr. Jon Wade)**

5.1. **Prototype Completion:** Completion of prototype technology such that it can be easily migrated and maintained.

5.2. **Migration, Open Source Hosting & Development, and Ticketing:** Migration and stabilization of the EA prototype to open source hosting and development site(s), and the support of a ticketing system and governance model to support its evolution.

5.3. **Tool Completion:** Completion of the tools to reduce the effort and technical skills of the development community. This is likely to involve the development of a new tool for editing experience phases and events. Please note that the tool for editing experience phases and events is not a deliverable for this task.

5.4. **Design Flow:** Creation and documentation of a design flow that can be used as an exemplar for external development.
5.5. **Documentation:** Documentation in a supportable open source format of the concept of operations, system, architecture, design, tools and content.

6. **External Developers Engagement:** This involves working with external development teams, likely academic and government researchers, on relative Experiences. These activities will both increase the content base and will assist in directing and validating the Open Source Preparation and Development work. This activity will be completed at Georgia Tech and Stevens Institute of Technology. **(Dr. Doug Bodner, Dr. George Kamberov, Dr. Richard Turner, Dr. Jon Wade)**

7. **Develop Multi-Learner Technology:** This research involves refining the current architecture such that it can effectively support multi-learner experiences which can be used to provide the ability to rapidly and inexpensively develop and iteratively improve and expand learning materials. This architecture should be developed such that it allows for the independent evolution of the constituent technologies, enabling long-term support and feature enhancement. In addition, this should provide an open source vehicle for distributed innovation.. This activity will be completed at Stevens Institute for Technology. (**Dr. George Kamberov, Dr. Jon Wade)**

8. **Write Final Report**: The findings from the above research activities will be summarized in a final report. The final report will describe the additional work that will be necessary to fully deploy the system. In addition, opportunities for expanding the use of the technology will be described for further development. These findings will be presented and reviewed with the sponsor and translated into a plan for the work in Increment Year 3. *(This activity will be completed by the entire project team, based at Stevens Institute of Technology, Purdue University and Georgia Tech.)*

1.3.2 Risk Management

In addition to the work activities, five top program risks were identified and tracked throughout the first year of the program in the following areas:

1. Project Management
2. Configuration Management
3. Technology Development
4. Content Development
5. Evaluation

Mitigation strategies were put in place as outlined in detail in the following sections.

The following is an updated version of the Risk Management plan for Increment 2.

1.3.3 RISK 1: PROJECT MANAGEMENT

Risk: Inability to support known and evolving customer/user feedback with current staff, budget and timeframe.
Mitigation: No significant new EA features are targeted for Increment 2, rather new capabilities will be restricted to those that address the feedback that we receive from learner evaluations of the Experience Accelerator. The work will be targeted at improving the current system to make it ready for Piloting.

1.3.4 RISK 2: CONFIGURATION MANAGEMENT

Risk: Inability to successfully manage the large number of files, configuration variables, present in the Experience Accelerator. (See LL4.2)
Mitigation: A more formalized approach will be used to provide assurance that the implemented configuration is in compliance with the desired specifications. This will be accomplished by creating a single design document that will be used with a work tracking tool to provide configuration management for the program. Reconciliation and updates in these two sources of data will be done on a weekly basis to ensure that control is maintained over the design.

1.3.5 RISK 3: TECHNOLOGY DEVELOPMENT

Risk: Inability to tradeoff long-term architecture and technology objectives (leading to successful open source support) vs. short-term prototype goals. One long term issue is the reliance on Flash which is a proprietary standard not being supported in some Apple client devices (e.g., iPad and iPhone). The likely standard to be supported in the future is HTML5. The time to make the transition will be dependent on when the toolkits are available to make it a productive environment and when it supported on browsers in use (the DoD used some old IE versions which do not support it).
Mitigation: Supporting this migration will require additional funding, with the timing of the migration likely to be post Increment 2. In the meantime, we will keep track of the evolving standards and attempt to reduce the impact of the eventual migration.

1.3.6 RISK 4: CONTENT DEVELOPMENT

Risk: Inability to produce a prototype that provides a compelling experience, supports the desired learning and is seen to be authentic.
Mitigation: Develop and review a design experience document which is used to guide the development process. This experience document will be improved to ensure that it contains the specific information necessary to facilitate configuration management. Unfortunately, due to the instability of the implementation in Increment 1, it was difficult to iteratively develop dialogue and feedback. However, the Experience Accelerator is now sufficiently robust that this

iterative approach can be taken. Additional tools will be explored that could improve this situation by providing the ability to quickly see the ramifications of specific learner behaviors.

1.3.7 Risk 5: Evaluation

Risk: Inconclusive results due to threats to validity of Experimental design (inability to generalize results), limited availability of suitable subjects and insufficient literature to support development of evaluation instruments. The critical challenge is to determine how to measure success in systems thinking and problem identification and resolution.
Mitigation: Additional work will be done to synthesize the published results in the literature. Explore development of new research instrumentation by synthesizing relevant literature should no suitable instrumentation be found in the literature. Create the capability to collect and analyze learner behavior traces, and compare pre- and post-experience traces of learners versus those of acknowledged experts. Possibly utilize Delphi sessions with SMEs will be explored as a means to develop a set of tests that can be used for pre- and post-Experience evaluation in these areas.

2 Prototype Development

The following describes the development of the Experience Accelerator prototype in RT16 Increment 2.

2.1 Development Process

At the beginning of the baseline year, given the exploratory nature of the research, activities tended involve communication of the entire research team. While this was effective in providing open communication between all of the team members, it was relatively inefficient and resulted in large parts of the team waiting for the completion of a particular piece of the program. Towards the end of the baseline year, the program migrated to a weekly meeting with the SMEs for content, a weekly team meeting and technology meetings on an as needed basis. In Increment 1, this was further refined to the development of a code and content Release Train in which major code releases were synchronized. The following are the three major code releases for Increment 1:

- **V1: Improve first-year prototype**
  - Stabilize operation
  - Complete implementation of existing features
  - Improve interfaces to facilitate updates and modifications
  - Update to conform with architecture

- **V2: Refine first-year prototype based on evaluation feedback**
  - Desktop usability improvements
  - Improved artifacts and dialog

- New dialog authoring tools and capabilities

- **V3: Add new features**
  - Earned Value Management (EVM) cost support
  - Variable reflection feedback to learner
  - Program actions feedback to learner
  - User interface additions and improvements
  - Preparation for open source support

In the Increment 2 year, releases were made on a more frequent basis dependent on the needs of the team for testing and evaluation, particularly with respect to external reviews with DAU instructors and students.

It was the intention in the baseline year for the team to develop technology and content using an agile software development process on a single server site at Purdue. However, it was found to be quite difficult to set up an iterative development environment and workflow. A major difficulty was in getting the necessary access for the team at the internal Purdue server site. Another challenge is that academic researchers are not necessarily full-time and it is very difficult to find frequent synchronization points. As a result we had the challenge of not having each of the development teaming working in the same environment with the same releases of code. We also had a rather adhoc version control system which consisted of uploads/downloads to/from DropBox.

In Increment 1, we have established an integrated development work at Stevens. In addition to this we have moved toward the use of a more robust configuration management and defect ticketing system, most of which will be used in an open source environment. The following are the major elements of this system:

- Software:
  - Source control: Subversion
  - Project Development, Management and Tracking: TRAC
  - Hosting: Stevens' server
- Content:
  - Dialog files – Chat Mapper
  - Integration: Dropbox, manual upload to Subversion
  - Software upgrades: versioned release trains with major and minor releases
- Documentation:
  - Move to Wiki-site at version 2.0

While the intention was to move to a wiki-site for all documentation, this migration did not yet taken place due to priorities being on the experience design and technology development. We have continued to use this approach throughout Increment 2.

Synchronization between the teams was still provided during weekly status meetings, and one to one conversations such that everyone has the opportunity to provide feedback including an assessment of how those changes might impact the work in their area going forward -- before the decision to adopt the change is made.

Another lesson learned in the baseline year is that the various pieces of the Experience Accelerator are closely linked and it is difficult to separate them. Significant effort was spent in Increment 1 to improve these interfaces such that artifacts and dialog can be created without the involvement of developers. This work was continued in Increment 2, which has allowed the teams to work rather independently. In addition, some tools have been created or procured that allows the content creators to test portions of what they are developing, e.g. dialog, as it is being developed.

2.2 DAU Course Integration

2.2.1 Overview

During Increment 2, the Experience Accelerator was targeted as a component in the DAU SYS302 Course through Exercise 5. To support this activity, the Experience Accelerator was modified to support the following Experience.

The goal is for each student to complete all five phases of the UAV experience twice: once as part of a multi-role team and the second individually. Due to time limitations, the teams will focus primarily on Phase 2, Preliminary Design Review (PDR) to Critical Design Review (CDR), working interactively with the instructor playing the role of the Program Manager. Exercise 5, which focuses on determining whether or not to proceed with the CDR based on the Experience Accelerator simulation. The remaining phases will be experienced as a team in an accelerated manner without instructor interaction. Once this is completed, the members of each team will complete Phase 7, Reflection with instructors available to answer questions and provide guidance. Students will then re-run the experience individually, acting as the Lead SE, at their own pace outside of class, also without instructor interaction. The class time allocated to this is the same as that for the current Exercise 5. Table 1 provides the schedule for this segment of SYS302.

Table 1: SYS 302 EA Deployment Schedule

Time	Instructors	Students
Tuesday		
1400-1430	Introduce Exercise	Team formation, role clarification/alignment
1430-1530	Mentor and Support	Individuals complete EA Phase 0 and Phase 1
1530-1630	EA	Team completes Phase 2A Cycle 1

	PM/Mentor/Control	
1630-1700	EA PM/Mentor/Control	Team completes Phase 2A Cycle 2
Wednesday		
0800-0830	EA PM/Mentor/Control	Team completes Phase 2A Cycle 3
0830-0900	EA PM/Mentor/Control	Team completes Phase 2A Cycle 4
0900-1000	Mentor and support	Presentation Development
1000-1100	View presentations & note items for reflection material	Teams deliver presentations
1100-1130	Monitor and support	Phase 2B (Receive & discuss CDR Results)
1130-1230	Lunch (develop reflection material?)	Lunch
1230-1315	Monitor and Support	Phase 3/Phase 4/Phase 5 speed through
1315-1430	Mentoring guidance	Individuals complete Phase 7 - Reflection
Homework	Review logged results after completion	Individuals replay experience

The following is a description of the participants, roles, document access and activities in the exercise:

Participants: Single (1) lead learner, multiple (1-5) subordinate learners, single (1) instructor.

Roles: Lead SE (lead learner), various other roles are to be determined for subordinate learners, PM (instructor), mentor (instructor).

Document, Artifact and Dialog Access: All roles have access to the same documents and may call or contact any of the NPCs.

Cycle activities: Each Phase 2 cycle follows the flow shown in Figure 2.

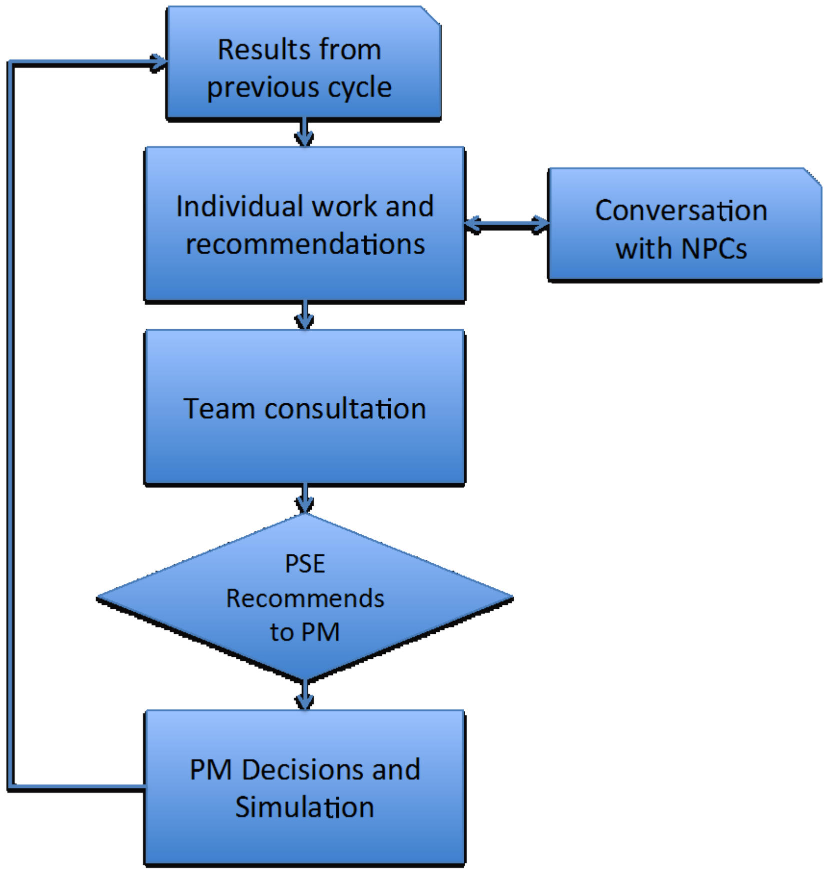


Figure 2. Generic EA Phase 2 Cycle Flow

Team Communication: All of the team members can chat or exchange email with one another. *Only the LSE can contact the PM*. The LSE and PM can have live chat and exchange email.

Non-LSE roles: All of the live roles can enter and submit their recommendations on a recommendation form. Their completed forms are placed in a common shared file that all can access.

LSE Role: The LSE determines when to submit the team recommendation form and subsequently when to request a cycle transition. The LSE can submit the team recommendation form to the PM *at any time*. This can happen regardless of whether or not the other players have submitted their forms (executive priority).

Decision Making: Only the LSE provides the PM with the team recommendations. However, since each team member's recommendations are logged, it is possible to review all suggestions in the reflection period.

Reflection: Each learner has a reflection period with common and individualized elements. All of the learners have access to the same feedback that the LSE receives regarding the success or failure of the project, recommendations made to the PM, and other information. All will receive feedback on the strength and/or weakness of their rationale for their recommendations. Reflection feedback will be automatically generated by the EA and can be presented to each of the learners. Once the reflection information is complete, access to the reflection period will be enabled for the learners.

The individualized elements are as follows:

- If the LSE makes recommendations that are not supported by the team that were incorrect:
    - The LSE will be asked why the team members who had the right idea were not more seriously considered.
    - The subordinate learners will be asked how they could have been more persuasive.
- Learners will need to report separately on lessons learned and how to apply their learning to the real world.

Instructor Activity: The instructor plays the PM and mentor role and also controls much of the action during the EA experience. There is a good deal of leeway available to allow for personal preference. The instructor participates in the experience directly as the Program Manager. Once the LSE has submitted his/her recommendation form, the PM is notified and can determine which of the recommendations to approve based on the LSE's rationale. The instructor enters his/her rationale for acceptance. This can be placed into a dialog for the LSE to query, or in an email (less realistic, but easier to implement). The instructor can pre-determine which recommendations to review and which to approve automatically (probably in an options file). This allows the experience to proceed without having the instructor in the critical path. The LSE will be notified in the main window when the next cycle's results are available

While the instructor in the pilot will not have control over the difficulty of the experience, it is anticipated that based on the feedback from the pilot, this capability will be developed. This can be based on personal knowledge and/or each individual's self-evaluation profile from Phase 0. In addition, the instructor can enable or disable "chance cards" such as landmines, bonuses, availability of additional information, etc.
All actions are logged for the lead and subordinate learners, especially their recommendation forms (which are already shared). The instructor can designate which of the following forms of post-processed logged information is desired:

- **Recommendations:** from LSE and subordinate learners for each cycle.
    - Note: This could be post-processed so that it is condensed into a single sheet to show the LSE recommendations for the entire experience, and show where the subordinate learners agreed or disagreed with the LSE's final conclusions. There could also be a form that shows which recommendations were accepted/rejected by the PM and why.
- **Dialog:** whom each learner contacted and which questions were asked.

- **Documents:** which learners read which documents and possibly how much time they spent on each document.
- **Activity:** when each learner logged in and out, total number of logins and time in each cycle, phase and experience.
- **Results:** final phase completed, project results (time to completion, money spent, range and quality), calculated score.

In the Reflections activity, the instructor will have the ability to add additional comments to this feedback, but, if desired, can preset an option for information to be automatically presented without comment.

2.2.2 NEW FEATURES AND CAPABILITIES

A number of new multiplayer capabilities were prototyped to support the integration of the Experience Accelerator into SYS302. These features are described in more detail in the Technology Development: New Features and Capabilities Section. In addition to this, a 42-page Instructor's Guide was created that supports the both instructors and students. The following are the contents of the guide:

Student Handouts

EA SYS302 Timeline	2
EA Recommendations Dashboard Scratch Sheets (5)	3
EA Student Reflection Questionnaire	8
EA Overview	9

Instructor Material

EA Introduction and Planning	15
Worksheet to connect EA to prior course activities	18
EA Facilitation Guide – Reference Sheet	21
Instructor Overview into SEEA Learner Choices	27

These documents evolved over the course of the various DAU instructor and student pilots and are expected to continue to evolve based on instructor and student feedback and need.

2.3 EXPERIENCE DESIGN

The following provides an overview of the experience design and the work that was accomplished in Increments 1 and 2. (For a detailed description of this work, see *The Experience Accelerator: Experience Design Document*.)

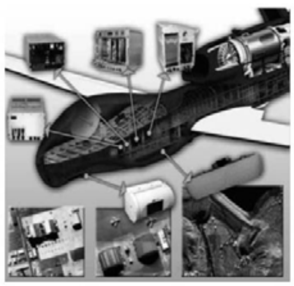
2.3.1 Overview

It is believed that accelerating the learning and maturation of Systems Engineers requires:

- Viewing a program through the entire lifecycle
- Seeing the relationships between elements of the system, and the system developing the system
- Encountering the challenges faced in a complex system development
- Being able to navigate through the “gray” zone
- Creating mental templates which can be applied to similar future situations

UAV System:

- S0 – System (UAV)
- S1 – Airframe and Propulsion (A&P)
- S2 – Command and Control (C&C)
- S3 – Ground Support (GS)



UAV KPMs:

- Schedule
- Quality
- Range
- Cost

Phases:

- EA Introduction
  - Phase 0 (P0): New Employee Orientation
- Experience Introduction
  - Phase 1 (P1): New Assignment Orientation
- Experience Body
  - Phase 2 (P2): Pre-integration system development -> CDR
  - Phase 3 (P3): Integration -> FRR
  - Phase 4 (P4): System Field Test -> PRR
  - Phase 5 (P5): Limited Production and Deployment
  - Phase 6 (P6): Experience End
- Experience Conclusion
  - Phase 6 (P6): Reflection
- Each session = 1 day

Figure 3: A Day in the Life of a LSE

The SE Experience initially developed in the baseline year focused on developing the systems thinking, and problem solving and recovery skills of a DoD Lead Systems Engineer. As shown in Figure 3, the SE Experience is designed to provide this learning by simulating the lifecycle of a UAV in which the learner is brought into the program after the Preliminary Design Review and is responsible for discovering the issues in the program and making the appropriate recommendations to correct the situation. The UAV system consists of three major subsystems of which the Airframe and Propulsion is primarily electro-mechanical, the Command and Control system is mainly software, and the Ground Support system is mainly human based. The major key performance measures (KPMs) are schedule, quality, range and cost, with cost being newly added in Increment 1. Each of the learner’s sessions in the Experience represent a single day in the program and are estimated to take approximately one hour to complete, although the learner is free to login and out any number of times during a session. The difficulty of the experience is determined by the learner's self-evaluation of their competencies and past history in the experience.

2.3.2 NEW FEATURES AND CAPABILITIES

A number of new features and capabilities have been suggested by the sponsors of this program, by the subject matter experts and by the RT16 design and development team. These changes were made in three primary areas:

- Dialog enhancement
- User interface and status visibility
- Instructor interaction in class environments

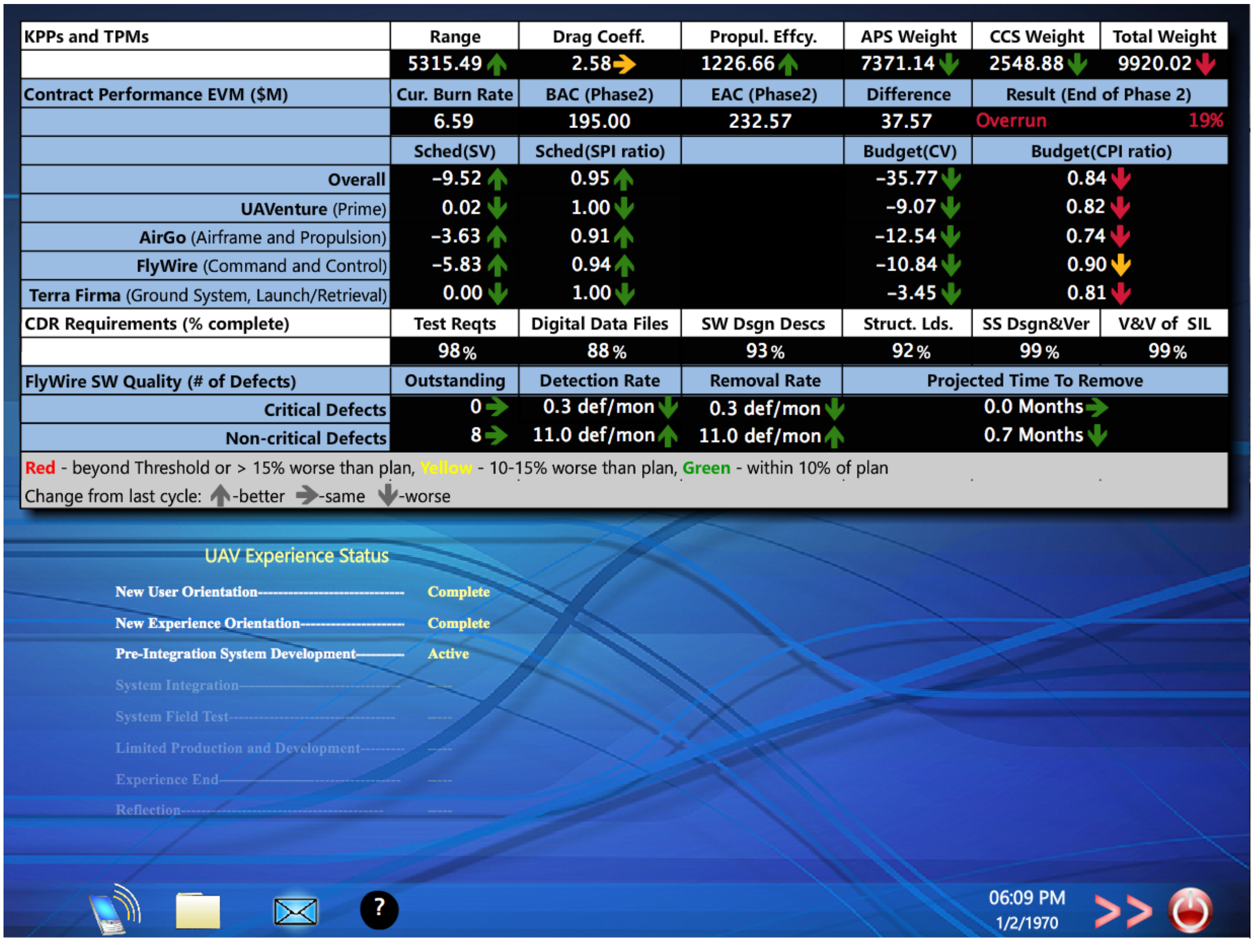
In addition to these new features, there has been significant work done in correcting errors and smoothing the learner and teacher activities.

2.3.2.1 User Interface and Status Visibility

The User Interface was revised in several ways. The primary change, however, were in the status display and the recommendation forms.

Status Display

One comment that was made from both Experience Accelerator users and people present at Experience Accelerator presentations was the difficulty in understanding the status based on the early meter display. It was decided that a different style of dashboard was needed that would provide status, values, and easy navigation to more specific information. A new dashboard was designed and revised during the classroom prototyping and is shown in Figure 4. It provides immediate access to current values, status, and trends for KPP, TPM, next-milestone-review task completion, and EVM data by contract. It also allows the learner direct access to the simulation-derived chart behind a specific value via clicking on the value. The dashboard for each phase will change to provide phase-specific information, although KPP, TPM, and EVM data will still be shown.



KPPs and TPMs	Range	Drag Coeff.	Propul. Effcy.	APS Weight	CCS Weight	Total Weight
	5315.49	2.58	1226.66	7371.14	2548.88	9920.02
Contract Performance EVM ($M)	Cur. Burn Rate	BAC (Phase2)	EAC (Phase2)	Difference	Result (End of Phase 2)	
	6.59	195.00	232.57	37.57	Overrun	19%
	Sched(SV)	Sched(SPI ratio)		Budget(CV)	Budget(CPI ratio)	
Overall	-9.52	0.95		-35.77	0.84	
UAVenture (Prime)	0.02	1.00		-9.07	0.82	
AirGo (Airframe and Propulsion)	-3.63	0.91		-12.54	0.74	
FlyWire (Command and Control)	-5.83	0.94		-10.84	0.90	
Terra Firma (Ground System, Launch/Retrieval)	0.00	1.00		-3.45	0.81	
CDR Requirements (% complete)	Test Reqts	Digital Data Files	SW Dsgn Descs	Struct. Lds.	SS Dsgn&Ver	V&V of SIL
	98%	88%	93%	92%	99%	99%
FlyWire SW Quality (# of Defects)	Outstanding	Detection Rate	Removal Rate	Projected Time To Remove		
Critical Defects	0	0.3 def/mon	0.3 def/mon	0.0 Months		
Non-critical Defects	8	11.0 def/mon	11.0 def/mon	0.7 Months		

Figure 4. New Dashboard Status Display (Phase 2a example)

Recommendation Forms

There was also feedback that the initial recommendation forms were not specific enough and the guidance for their use was difficult to follow. The forms were revised and an example is shown in Figure 5. The forms now change by Phase, and only recommendations available for the current phase are shown. The number of recommendation areas has also been revised so that changes outside of the LSE are not available. For example, changing the value of a KPP is probably not within the purview of the LSE, but could be included as a point of discussion.

For all changeable parameters, the original, current, and target values are shown. Parameters often take time to actually reach the recommended levels, so this better demonstrates the lags in personnel changes or the time it takes to see actual technical progress through TPMs.

One significant issue raised in the DAU pilot was the necessity to explain why the Lead Systems Engineer would have the opportunity to make personnel resource allocation recommendations. While it is inappropriate to and actually illegal for the LSE in a government program to make recommendations like this, they are being used as surrogates as a teaching tool to increase the learner's understanding of system development issues.

2.3.2.2 Dialogue Enhancements

Because of development schedules and rapid prototype feasibility requirements, the initial dialogues tended to be based on a single thread of actions on the part of the learner, and only varied slightly from cycle to cycle. As the work evolved, the dialogues became richer. They became more frequent within the experience, the NPCs gained more developed personalities and provided additional clues, and the dialogues reacted more to the simulation and evolving conditions of the story.

Temporality

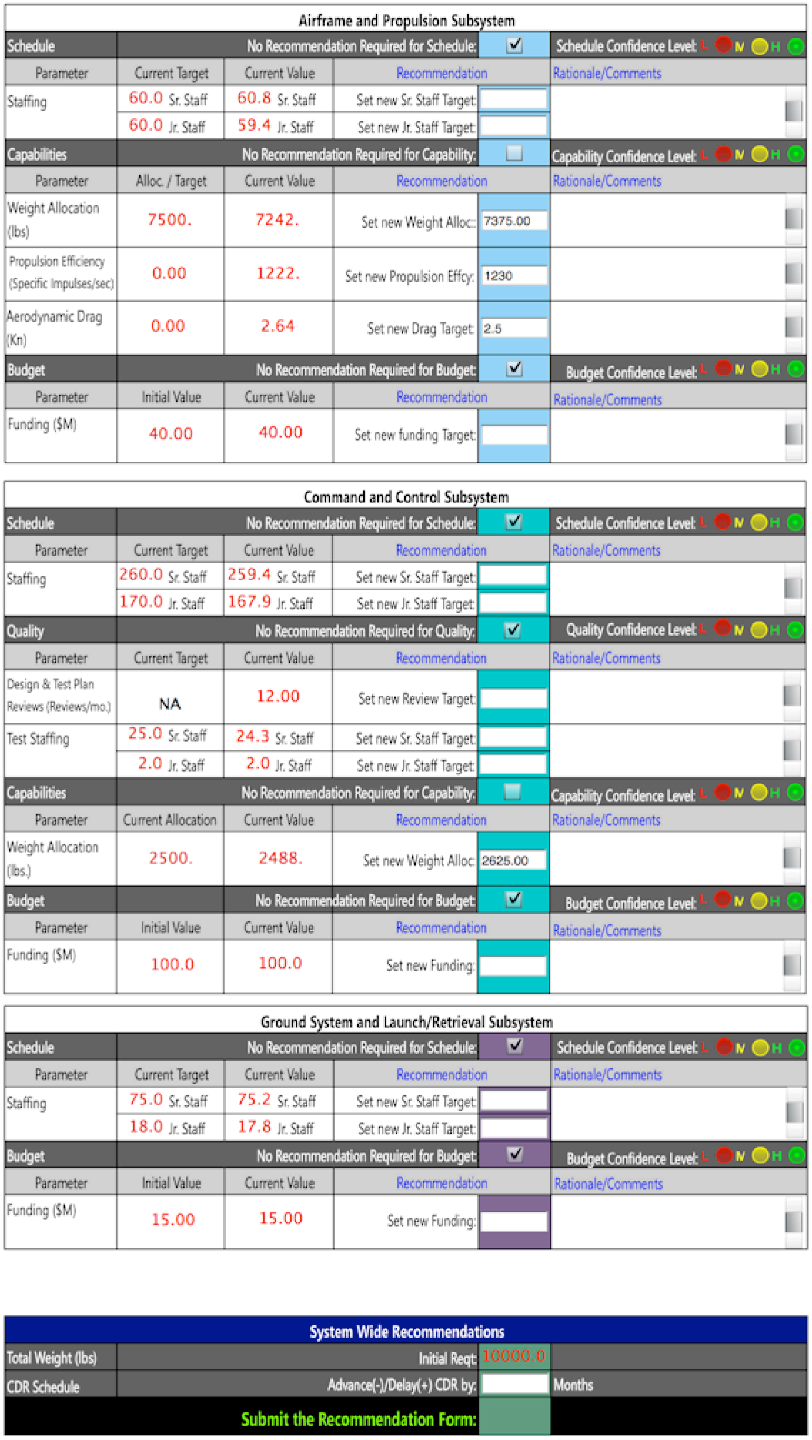
The dialogues had primarily been developed on a cycle basis rather than a sub-cycle basis. However, it was soon apparent that each sub-cycle, with the learner's recommendations and the simulation results based on those recommendations, was the de facto cadence. To maintain the information flow to support the learner, dialogues were developed for each subcycle. This involved tracking the current subcycle status and providing different conversation tree branches based solely on the temporal subcycle. This allowed dialogues to be based on time-critical issues and to adjust if recommendations about delays were made. It also allowed insertion of challenges based on external events (such as unexpected changes to requirements, resources or schedules).

Relevance

The dialogues also needed to better track the information that was provided in the recommendations of the learner and the results they generated in the simulation. To support this, dialogue variables were provided for conditional branching in dialogues. This allowed the dialogue to change based on the recommendations and simulation results. The NPCs can now provide additional information based on the simulation as well as the story line. The dialogues can also now display subject values to the learner during the conversation. While not necessary because of the new dashboard and recommendations interfaces, being able to use values in the conversation adds to its verisimilitude.

2.3.2.3 Instructor interaction in Class Environments

Initially, the EA was considered to be a stand-alone experience. However, the DAU classroom environment led us to investigate other modes. Of particular interest were different ways for the instructor to interact with the student learning teams.



Airframe and Propulsion Subsystem				
Schedule		No Recommendation Required for Schedule:	☑	Schedule Confidence Level: L M H
Parameter	Current Target	Current Value	Recommendation	Rationale/Comments
Staffing	60.0 Sr. Staff	60.8 Sr. Staff	Set new Sr. Staff Target:	
	60.0 Jr. Staff	59.4 Jr. Staff	Set new Jr. Staff Target:	
Capabilities		No Recommendation Required for Capability:	☐	Capability Confidence Level: L M H
Parameter	Alloc. / Target	Current Value	Recommendation	Rationale/Comments
Weight Allocation (lbs)	7500.	7242.	Set new Weight Alloc.: 7375.00	
Propulsion Efficiency (Specific Impulses/sec)	0.00	1222.	Set new Propulsion Effcy: 1230	
Aerodynamic Drag (Kn)	0.00	2.64	Set new Drag Target: 2.5	
Budget		No Recommendation Required for Budget:	☑	Budget Confidence Level: L M H
Parameter	Initial Value	Current Value	Recommendation	Rationale/Comments
Funding ($M)	40.00	40.00	Set new funding Target:	

Command and Control Subsystem				
Schedule		No Recommendation Required for Schedule:	☑	Schedule Confidence Level: L M H
Parameter	Current Target	Current Value	Recommendation	Rationale/Comments
Staffing	260.0 Sr. Staff	259.4 Sr. Staff	Set new Sr. Staff Target:	
	170.0 Jr. Staff	167.9 Jr. Staff	Set new Jr. Staff Target:	
Quality		No Recommendation Required for Quality:	☑	Quality Confidence Level: L M H
Parameter	Current Target	Current Value	Recommendation	Rationale/Comments
Design & Test Plan Reviews (Reviews/mo.)	NA	12.00	Set new Review Target:	
Test Staffing	25.0 Sr. Staff	24.3 Sr. Staff	Set new Sr. Staff Target:	
	2.0 Jr. Staff	2.0 Jr. Staff	Set new Jr. Staff Target:	
Capabilities		No Recommendation Required for Capability:	☐	Capability Confidence Level: L M H
Parameter	Current Allocation	Current Value	Recommendation	Rationale/Comments
Weight Allocation (lbs.)	2500.	2488.	Set new Weight Alloc: 2625.00	
Budget		No Recommendation Required for Budget:	☑	Budget Confidence Level: L M H
Parameter	Initial Value	Current Value	Recommendation	Rationale/Comments
Funding ($M)	100.0	100.0	Set new Funding:	

Ground System and Launch/Retrieval Subsystem				
Schedule		No Recommendation Required for Schedule:	☑	Schedule Confidence Level: L M H
Parameter	Current Target	Current Value	Recommendation	Rationale/Comments
Staffing	75.0 Sr. Staff	75.2 Sr. Staff	Set new Sr. Staff Target:	
	18.0 Jr. Staff	17.8 Jr. Staff	Set new Jr. Staff Target:	
Budget		No Recommendation Required for Budget:	☑	Budget Confidence Level: L M H
Parameter	Initial Value	Current Value	Recommendation	Rationale/Comments
Funding ($M)	15.00	15.00	Set new Funding:	

System Wide Recommendations			
Total Weight (lbs)	Initial Reqt:	10000.0	
CDR Schedule	Advance(-)/Delay(+) CDR by:		Months
	Submit the Recommendation Form:		

Figure 5. New Recommendation Form Display (Phase 2a example)

Three specific ways of instructor interaction emerged:

1. As a mentor without system interaction
2. As a passive Program Manager
3. As an active Program Manager

Mentor

This is the role that ended up being closest to the DAU traditional instructor role and was used in the pilot. It includes having the instructor watch over the student teams as they work. The instructor can provide guidance or answer questions, but is not directly involved in the scenario.

Passive Program Manager

This is an extension of the Mentor role with an ability to accept or reject the recommendations by the team.

Active Program Manager

This is the expected role for multi-player supported experiences (see section 2.4.2.1). The instructor replaces the system NPC PM as an active player with the Program Manager's role. It is up to the instructor to receive and respond to the recommendations of the teams. Tools are provided that allow the instructor to view all of the experiences of the members of the team, communicate via chat, and accept or reject the team's recommendations.

2.4 TECHNOLOGY DEVELOPMENT

2.4.1 OVERVIEW

The EA game engine has two components: the runtime engine and the tools suite. The tools suite includes the Experience Development and Simulation Engine-related tools described in the Section 2.6.

In this section we will describe the EA Run Time Engine component (EARTE). As shown in Figure 6 the EARTE has a layered architecture client-server incorporating the following modules:

- **Experience Master:** contains the overall Experience state and provides control and sequencing for the other major EA modules.
- **Challenge Control:** contains the Learner profiles and Experience history logs and leverages these in conjunction with the competency taxonomy and 'Aha' moments to determine the appropriate challenges and landmines for each Learner.
- **Simulation Engine:** determines the future state of the system and outputs to be presented to the Learner.

- **Non-Player Characters (NPC) Engine:** represents other non-player characters in the simulation, and creates and assembles the content for Learner interactions.

- **Presentation Engine:** accepts inputs from the Learner and provides the presentation of the Experience interface to the Learner.

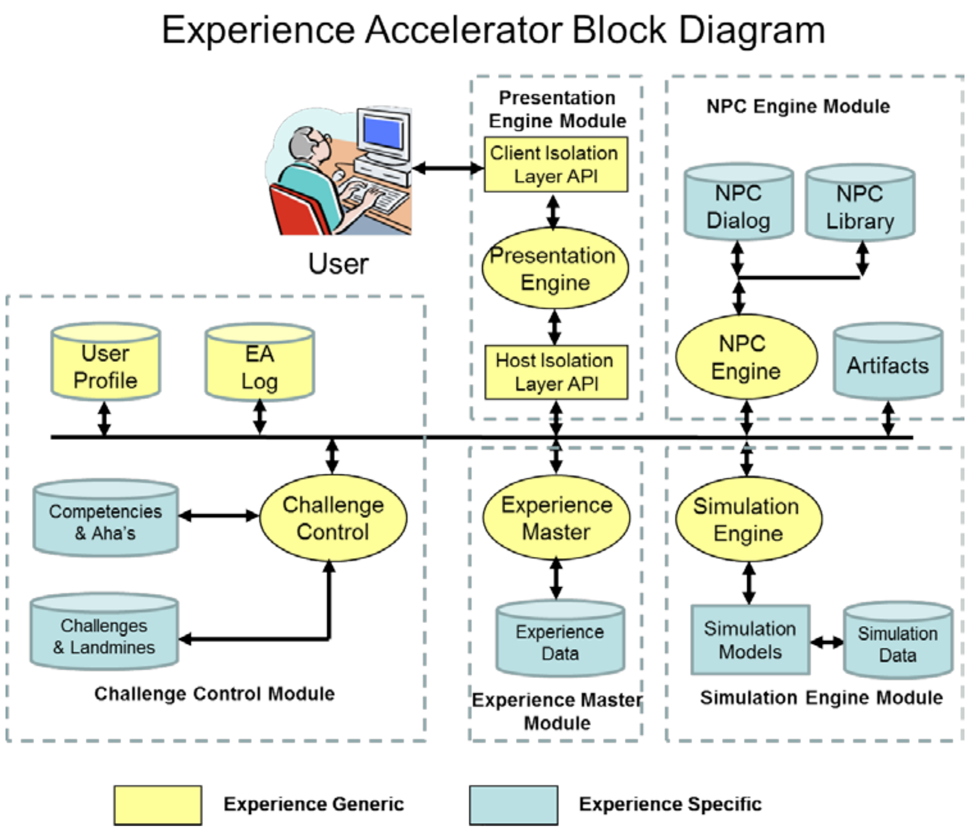


Figure 6: Experience Accelerator Logical Block Diagram

The EARTE is a multiuser architecture for internet gaming. It has light clients (currently implemented in FLASH) and a Java server which runs multiple game instances concurrently. In order to support as wide as possible range of EA games and scenarios the EARTE does not incorporate a simulation engine, but rather the NPC engine provides a framework to interface with 3d-party simulation engines. For more on the simulation see Section 2.5.

During Increment 2 the technology development team continued to refine the EARTE to achieve architecture adherence and improved runtime performance stability. The major new features and capabilities added in Increment 2 were the multi-learner capabilities, and redesign of the Dashboard and Recommendation forms. The following section provides an overview of the work that arose from suggested enhancements, fixes, and multi-learner capabilities.

2.4.2 NEW FEATURE AND CAPABILITIES

2.3.2.1 Multi-Learner

Prototype multi-learner capabilities have been developed for the Experience Accelerator. The capabilities have been provided for multiple learners to create and join games asynchronously, share documents, communicate directly with one another, and make decisions that affect the outcome of the simulations. The following are some of the possible modes that may be supported:

1. Single Learner mode
2. Single Learner with supervisor (PM & Mentor) [High]
3. Multiple Learner [High]
4. Multiple Learner with supervisor [High]

The following are some of the capabilities of the system:

1. Interface to create/join multi-learner games
   a. Participants join independently, each picking a role.
   b. When a participant selects multi-learner mode, a new screen type appears and the participant can determine if he/she wanted to join an existing game or start a new game. If starting a new game, the participant will create the name for that instance of the game. Once a participant has elected to start/join a game, he/she selects his/her role and then is moved to another waiting screen. Once a game is subscribed, it may start.
   c. Participants need to arrange a common time to play the game outside of the game.
   d. If a participant quits, then the role is vacant and someone else has to take the role before the game can proceed.
   e. The game can proceed if all of the “essential” players are there. The definition of “essential” is determined by the game and particular phase that is active. A future option would be to allow for a NPC in the place of a non-essential player.

2. The following are the different roles that are supported:
   a. Main learner
   b. Peer learners
   c. Instructor/Mentor (special setup & control options, e.g., degree of difficulty)

3. Artifact Access:
   a. Documents: are defined as being common (access to all) and role specific (only certain roles can access them)
   b. Document sharing is through a common directory, not through email or sending

4. Email: (not implemented)
   a. This will work like normal email in the experience

5. Text Chat:
    a. Works the same as email, but does not log information

6. Voice/video: (not implemented)
    a. Voice supported through a peer-to-peer, free application (Cirrus, Adobe, not open source). Nothing is required on the client. Video is also supported with this application. Conversations are not logged.

7. Decision Making
    a. During each phase, certain roles have defined decision making capabilities. In particular, this is the ability to fill out the recommendation forms for simulation.

8. Phase Transitions
    a. If the supervisor is present, he/she determines when to make a phase transition.
    b. If there is no supervisor, then the decision making learner makes this determination.
    c. In all cases, the other learners can provide their recommendation for when they want to change phases.

The critical multi-player features have been designed, implemented and tested to support DAU SYS302. However, there are known defects and instabilities that need to be remedied before this can be deployed in the classroom.

2.3.2.3 Dialog System

Additional capabilities have been enabled with ChatMapper output files such that conditionals are now supported that enables the dialog to be selected based on the state of the simulation.

2.3.2.3 Dashboard

A new dashboard was designed and implemented which provides a broad range of project information to the learner. Detailed status charts are directly linked to each dashboard metric so that the learner can quickly access the simulation results that are related to the dashboard metric. This is described in more detail in Section 2.3.

2.3.2.4 Persistence Support

Learner progress is more thoroughly recorded and logged allowing a learner to continue where they left off after logging out and returning at a later date. Persisted information includes:

- State information (schedule, budget, quality, etc.)
- Events such as emails and voicemails
- Learner input in forms
- Feedback to the learner in status update emails, and reflection documents

Stability issues have been addressed in this area in multi-player mode, by replicating this information for each client such that failure in one client does not impede the other clients from accessing the information.

2.3.2.6 Network Improvements

The client/server communication system continued to be improved in Increment 2. Numerous improvements were made in the architecture and design to increase efficiency and improve stability. Improvements were also made in the communication channel with the simulation which has greatly improved stability as well.

2.5 SIMULATION ENGINE

The following provides an overview of the simulator and the work that was accomplished in Increment 2.

2.5.1 OVERVIEW

The simulator module maintains the quantitative state of the program and advances this state in between learner cycles, based on changes that the learner might make to program parameters and variables. The simulator uses the well-known paradigm of system dynamics (Sterman, 2000), which allows the modeling of system behaviors over time featuring flows, lags, feedback loops and other non-linear phenomena. System dynamics has been employed in modeling and simulating systems engineering processes, most notably those involved in software development (Madachy, 2008). The simulator module contains the following components:

- The *execution engine* advances the program state via simulation and records values of updated program variables over time. It returns these values to the experience master, which can use them for scoring. It also generates output charts for the learner reflecting updated program status and progress. It maintains state variables in between simulation cycles and phases via various files, and it also reads from files provided by the experience master that correspond to learner decisions about which program parameters or variables should be changed. It consists of Java-based, open source code based on pre-existing products.
- Various *simulation models* specify program parameters, state variables, their behavior over time, and their interactions. These are implemented as XML files and are executed by the execution engine.
- *Output charts* provide graphs of program state variables over time reflecting program status, behavior and indicators. These charts are in the form of graphics files fed to the experience master based on an XML chart specification. The specification defines the chart format, which is then filled in by values of program variables over time.

A simulation model specifies the behavior over time of a system of interest in terms of:

- Variables – quantities that change over time during model execution,
- Parameters – quantities that do not change over time during model execution (but that can be changed via user action),
- Rates – the rates at which a variable changes (which may in turn change over time), and
- Auxiliary relationships – equations used to model the relationships between parameters, variables and rates.

These are expressed via an XML-based specification. A simulation model is provided for each of the different phases of the experience. Each model has a variety of sub-models that represent important elements of the acquisition program in terms of the engineering workforce and its functional areas (e.g., design versus quality assurance), cost accruals and budgets, defects (resolved, unresolved and unidentified), UAV system performance estimates, and progress toward entrance criteria for critical program reviews at the end of each phase.

Of course, there is significant interaction between the sub-models. For instance, the workforce accrues labor cost over time. This may be at the scheduled accrual rate, or it may be at a higher accrual rate, especially if the schedule is falling behind. The productivity of the workforce impacts the progress toward meeting entrance criteria for reviews. Performance measure estimates for the UAV system often degrade over time, unless workforce resources are called to address problems. Below is a summary of the various sub-models in each phase.

- Phase-2 – Pre-integration
    - The UAV sub-model tracks estimated range plus underlying estimates of technical performance measures such as drag, propulsion efficiency, overall weight and sub-system weight.
    - The design workforce sub-model converts full-time equivalent engineering headcounts (at two levels of experience) to progress on various tasks using productivity, training and communication overhead effects. There is a design workforce for each of the sub-system contractors plus the prime contractor.
    - The entrance criteria sub-model represents the progress over time for the various entrance criteria needed for Critical Design Review (CDR). The labor sub-model results feed this progress. Each contractor's workforce provides a contribution to progress on each criterion, depending on its involvement. Thus, each workforce typically splits its results among entrance criteria.
    - The quality sub-model tracks the number of defects encountered over time plus the workforce that addresses these defects. Defects must first be identified and then resolved. Design reviews can help identify defects. It should be noted that only software quality for the command-and-control system is currently modeled.
    - The EVM sub-model is used to track the accrual of costs over time and compares these to the expected costs over time. Thus, it can be determined where the program is relative to schedule and budget. There is an EVM sub-model for each

contractor, including the prime contractor, and these roll up into an overall EVM sub-model.

- Phase-3 – System integration
    - The UAV sub-model operates quite similarly to Phase-2.
    - The design workforce sub-model is similar to that of Phase-2 except that the prime contractor is more heavily involved due to the integration nature of the work.
    - The entrance criteria sub-model is similar to that of Phase-2, except that the specific criteria are for Flight Readiness Review (FRR), and the contributions of each workforce are consequently different.
    - The quality sub-model operates similarly to that of Phase-2. Once again, only software quality for the command-and-control system is currently modeled.
    - Finally, the EVM sub-model operates similarly to the sub-model of Phase-2. The budget and cost parameters are, of course, different.

- Phase-4 – Flight test
    - The UAV sub-model operates quite similarly to Phase-2 and Phase-3.
    - The workforce sub-model continues to operate similarly to that of Phase-2 and Phase-3, except that it is addressing test rather than design.
    - The entrance criteria sub-model is similar to that of Phase-2 and Phase-3, except that the specific criteria are for Production Readiness Review (PRR), and the contributions of each workforce are consequently different.
    - The quality sub-model operates similarly to those of Phase-2 and Phase-3. Of course, during flight test, it is expected that there are fewer quality issues than before.
    - Finally, the EVM sub-model operates similarly to the sub-models of Phase-2 and Phase-3. The budget and cost parameters are, of course, different.

- Phase-5 – Low rate initial production (LRIP)
    - The UAV sub-model operates quite similarly to previous phases.
    - Instead of a workforce model, there is a production model. Incremental progress on production of each air vehicle sub-system (discretized into release of finished sub-systems). These are then available for integration into an air vehicle. Thus, it is desired that the production rates of each air vehicle sub-system (airframe and command-and-control) are roughly equal over time. Ground stations are produced separately. Finished air vehicles and ground station units are then compared against the production target for each in this LRIP.
    - There is no entrance criteria sub-model.
    - There is no quality sub-model, although one could be introduced for production defect issues.
    - Finally, the earned EVM sub-model operates similarly to the sub-model of previous phases. The budget and cost parameters are, of course, different.

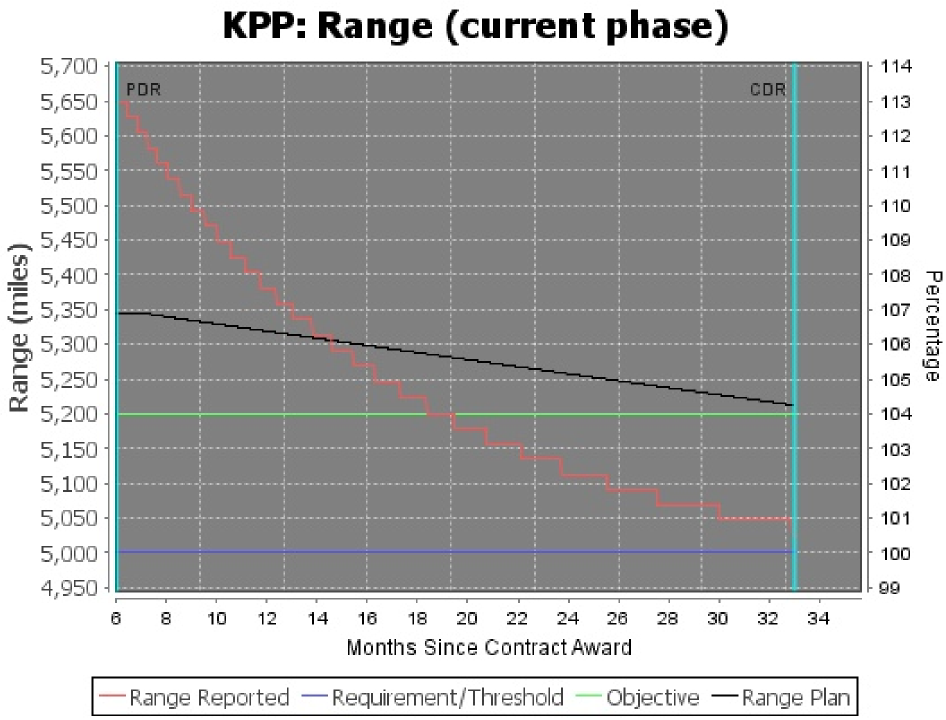
A set of text files maintains state variable values between phases. Thus, the variable for drag is written to a text file at the end of each cycle within Phase 2. Once Phase 2 terminates, the current value for drag is written to the Phase 3 model.

The simulator provides output artifact charts so that the leaner can see the results of his/her decisions integrated with the continued progress of the program. These charts fall into the following categories.

- Acquisition program baseline (APB) metrics. These metrics track program performance related to baseline plans as the program moves from design and development to production.
    - Key performance parameters and technical performance measures (KPPs/TPMs). These charts track estimates and actual for metrics related to performance on program requirements.
    - Other important system attributes. These metrics track non-requirements performance, typically relate to production cost per unit, maintenance costs, training costs, operational personnel required. These are tracked as the program moves from design and development to production.
- Quality. These charts track key indicators of quality apart from program requirements.
- Review meeting entrance criteria. These charts track progress on meeting the various entrance criteria associated with review meetings. They serve as indicators of program schedule.
- Cost and budget. These charts track cost and schedule performance relative to budget using the concept of EVM. EVM utilizes a number of metrics over time to track performance.
- Initial operating capability (IOC) metrics. These charts track system production and delivery status in terms of meeting IOC (number of systems to be delivered by scheduled delivery date).

As shown in Figure 7, a particular chart typically has values over time of:

- One or more metrics of interest,
- A target or historical benchmark for each metric, and
- A plan for where each metric is expected to be given expected program progress and performance.



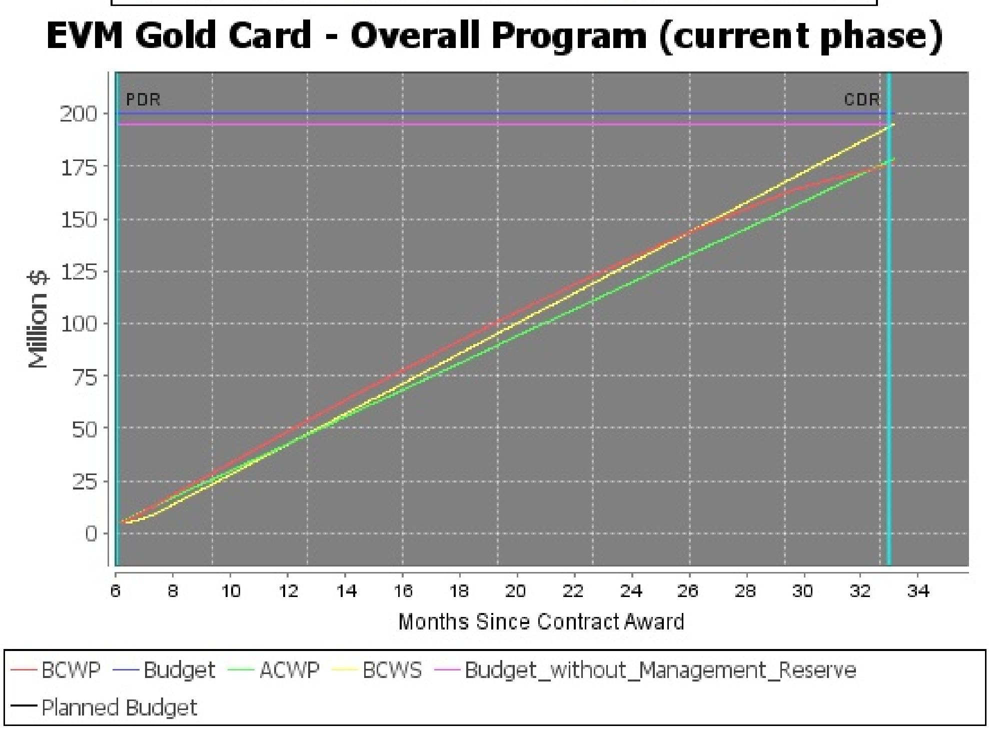


Figure 7: Example Charts

2.5.2 NEW FEATURES AND CAPABILITIES

A number of new simulator features and capabilities were implemented this past year. These resulted from continued progress plus input from subject matter experts on program behavior and output chart realism and from DAU instructors on concepts being taught in their curriculum. This section summarizes this work by simulator module component. Note that many improvements have interdependencies between components (e.g., changes to chart format require changes to execution code needed to generate new formats).

2.5.2.1 Execution Engine

The execution engine was extended in a number of areas, mostly to facilitate desired model and chart features.

- The code was enhanced to support a number of new features of the simulation models and output charts.
- The code was tested to determine support for higher resolution charts. This can be supported, but the experience master needs to be updated to support this, and a specific resolution needs to be defined.

2.5.2.2 Simulation Models

New features were developed for the various simulation models, with a focus on the Phase-2 model since the pilot usage at Defense Acquisition University was set to use only this part of the learning experience.

- Support for minimum and maximum values of simulation variables was added in the model specification syntax. Minimum and maximum values were implemented for such variables as entrance criteria progress (e.g., range between 0% and 100%).
- Support for non-increasing and non-decreasing variables was added.
- The software error variables were discretized for the command & control sub-system.
- The Phase-2 model was adjusted so that the program is fully staffed at the beginning of the experience. In addition, the experience mix in the various engineering workforces was adjusted per SME input. A “standard” workforce for a program is 80% experienced and 20% inexperienced. A lead systems engineer would note a red flag if the experience levels were weighted with more inexperienced staff. Thus, the airframe & propulsion and the command & control workforces were set to have 50% and 35% inexperienced staff, as an indicator that there would probably be productivity issues with these two sub-systems.
- The productivity differences between experienced and inexperienced staff were increased.
- Target variables were added for the various workforces and TPMs. The learner can set a target variable in his or her recommendations, and the simulation will move the value of the variable toward the target. In doing so, there is a delay between the target setting and target achievement. In addition, the variable may have a minimum or maximum

value, thus affecting whether the learner can achieve the target. Finally, moving the target value may cause the simulation to apply resources (e.g., staff or cost) to represent the assignment of resources to meet a goal.

- For drag and propulsion efficiency targets, the simulation engages senior staff from the airframe & propulsion contractor based on the difference between the actual TPM and the target. The greater the deficiency, the more staff are applied to fix the problem.
- For the weight allocations, the simulation applies senior staff from each of the relevant sub-system contractors based on the difference between the allocation and the actual weight. The closer the actual is to the allocation, the fewer staff resources are applied. Similarly more staff resources are applied if the actual is over the allocation.
- A design objective was added to range. This is in addition to the existing plan and minimum requirement.
- A set of conditions for CDR success was designed (e.g., range above design objective and all entrance criteria above 90% complete).
- The Phase-2 model was tested extensively for correct operation.
- An Excel-based prototype tool was developed to aid in testing and tuning.
- The Phase-2 simulation model was tuned by DAU request to have a scenario where, with the “optimal” learner inputs, the cost overrun would be approximately 20% with a three month schedule slippage. This represents a successful outcome, given the problems inherited by the learner at the beginning of the experience.

2.5.2.3 Simulation Output Specification

The simulation chart specification syntax was modified to support several new features.

- Percentage values for y-axis variables if desired. These are displayed on the right-hand side of the chart, whereas the numeric values are displayed on the left-hand side. The percentages are relative to some requirement or target.
- Step function values for entrance criteria to support criteria that consist of several sequential sub-tasks. In addition, labels for the sub-tasks can be displayed on the chart.
- Specification for program-length charts versus the phase-length charts previously supported.

Currently, the chart specification for a particular simulation model resides in a companion file to that model’s XML file.

2.5.2.4 Output Chart Artifacts

A number of new charts and chart enhancements were added this past year. A summary is included below.

- Plans were revised for all weight charts to reflect historical weight growth based on SME input. The range plan was adjusted accordingly.
- Two sets of charts are now provided. The first set reflects the values of the variables and plans graphed during the particular phase in which the learner currently resides.

These are displayed via click-throughs on the newly implemented program dashboard, and were previously available in the output chart slide deck. New charts were added with extended plans and variables over the program life. These are now available in the slide deck.

- Burn rate charts were added for each of the sub-contractors, for the prime and for the overall program. These are based on monthly expenditure rates. Planned rates were incorporated based on the respective budget.
- Personnel FTE (full-time equivalent) level charts were added for each of the sub-contractors, for the prime and for the overall program. This allows the learner to track staffing over time.

2.5.2.5 Support for Other EA Modules and Documentation.

The following support was provided for other modules in the Experience Accelerator, as well as documentation for instructor usage.

- Simulation variables were mapped to the various values displayed on the dashboard, as well as plan/target values for those variables so that their status relative to plan/target (i.e., green, yellow, red) can be displayed. These are also used by the NPC module for state-based dialog.
- The mapping of simulation variables to recommendation form items was modified so that the learner sets target for various items (e.g., staffing or TPMs), and then these are written to the simulation's corresponding target variables.
- An overview write-up of the simulator and simulation models was provided for the instructor packet.
- An analysis was performed comparing a "do-nothing" scenario where the learner takes no action to an "optimal" scenario, where the learner takes recommended actions. This was written up for the instructor packet.

2.6 EXPERIENCE DEVELOPMENT TOOLS

2.6.1 FUTURE TOOLS

The following are a number of candidates for tool development that could have a very significant impact on content creation productivity. A subset of these tools will be targeted for development supported by SERC funding.

2.6.1.1 Phase and Event Specification

Currently the specification of Experience phases and events is one where the content creator specifies this is text, uses a simple format as noted in the Experience Design document. Once this is completed, the technical developers must read this information, interpret it unambiguously and then change the appropriate source code in the Experience Master modules. As one may imagine, this is a tedious process that is error prone and difficult to debug during actual Experience operation. An alternative to this approach has been designed in concept that involves the creation of a graphical interface based tool that allows the content creator to specify the desired Experience phases and events in a natural way similar to the diagram shown in Figure 8.

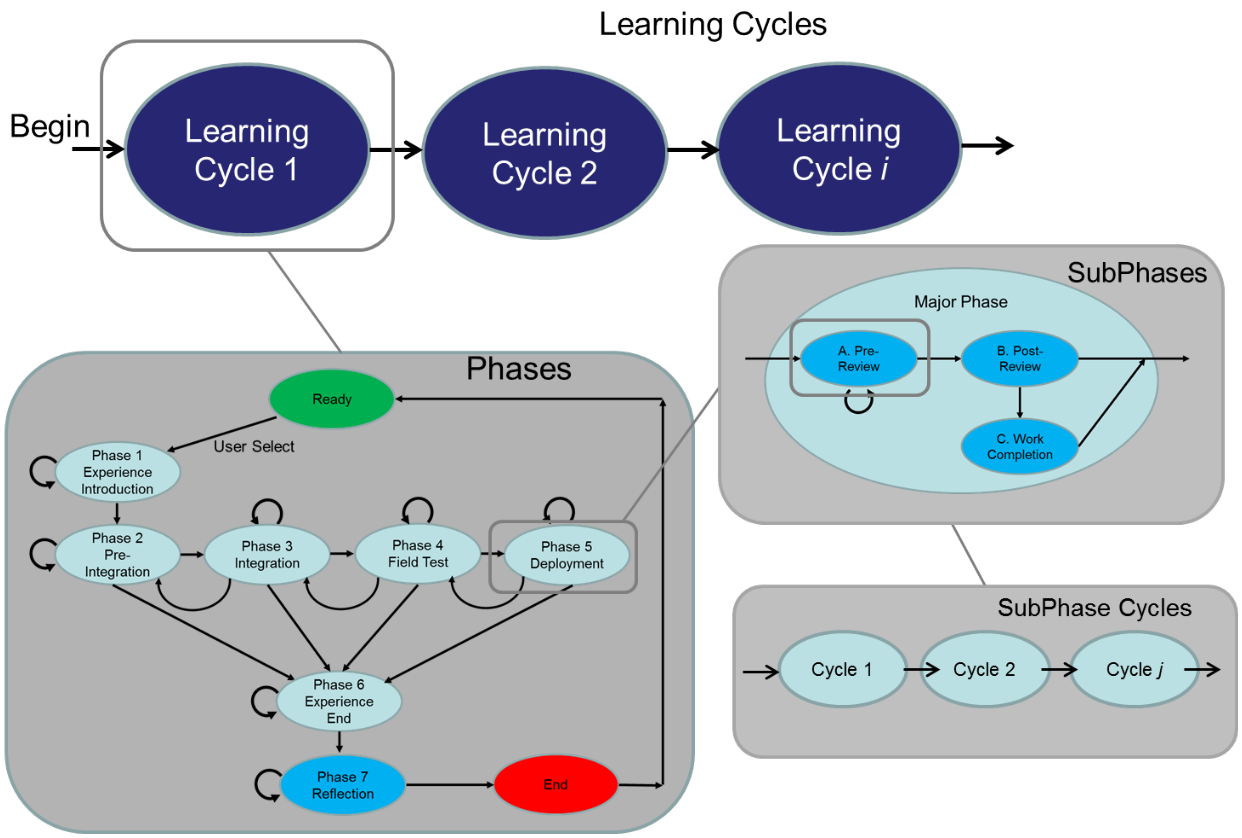


Figure 8: Experience Learning Cycle and Phases

2.6.1.2 Artifact Entry

Another tedious, cumbersome and error-prone process is that of artifact entry. In the current Experience, there are hundreds of artifacts that may be updated within a development cycle. As noted below, currently there are half a dozen steps in the process that involve both the content designer and the technical developer. Requiring the technical developers to get involved in the process greatly slows down the content development work, particularly in an academic or open source environment as there is not full-time staff standing by to assist the content developer. A conceptual design has been done of the development of an application that the content designer could use to eliminate the interaction with the technical developer to reduce the total effort, reduce the time delay in making changes and finally reduce the errors in the process.

- Today:
  - Designer saves file in DropBox
  - Designer tells technical staff to load it into the design

- Technical staff moves file to the correct location or handcodes changes
  - Technical staff recompiles or links the new code
  - Designer is notified of the change
  - Designer tests changes
- Future:
  - Designer opens artifact entry client
  - Designer saves file into system sandbox
  - Designer tests changes

2.6.1.3 Simulation Model Construction and Parameter Setting

One of the most challenging and technical areas of the Experience content development work is in the construction of the simulation models that are used to provide a simulated experience to the learner. Effort is necessary both to develop the models (in the case of the current Experience these are case systems dynamic models) and to tune the parameters properly to ensure that their output is both plausible and provides the desired experience. Currently, existing models can be used as templates, and a library of such models can be constructed. Longer term, however, it is believed that there is an opportunity to development tools that can assist in the design and tuning process.

There is currently an application with a graphical user interface that can be used to develop single models. This application was enhanced during Increment 2 to support model features that were developed in Increments 1 and 2. In addition, Excel was used to prototype models and aid in tuning. The following are three major areas of potential tool development:

- Model Builder: construct models based on templates
- Parameter Tuner: highlight parameters with greatest impact over selected time scales
- Simulation Master: ability to run batches of simulations to accelerate tuning and validation of models.

2.6.1.4 Process and Tools Documentation

An outline of the steps needed to create an experience was created in Increment 2. This will need to be enhanced with detailed instructions for using tools to be developed in the future. In addition, external developers will be engaged to validate these processes and also to provide feedback necessary to improve and update these tools and environments.

2.6.1.5 Experience Learning Evaluation

Learner experience behavior trace and analysis tools should be developed to capture and analyze learner outputs so that the behaviors and decisions of learners can be compared with the behaviors of acknowledged experts in the areas identified to be stress in this experience. In addition, the amount of learning should be amenable for analysis by such tools. This effort can leverage the work that is being done at Stevens in other serious game environments.

2.7 PERFORMANCE EVALUATION

The performance of the Experience Accelerator was analyzed to determine the requirements for supporting deployment with the DAU with 30 active concurrent users. In addition, a number of recommendations were made for future improvements in execution efficiency.

2.7.1 ANALYSIS

Based on a review of the response latencies, it is believed that the simulation, file update and status chart generation activities provide the greatest resource load on the Experience Accelerator server. A series of profiling experiments on independent runs of the system dynamics simulation were conducted in order to determine the system requirements necessary to support 30 simultaneous Experience Accelerator users, which is the objective for the initial DAU deployment.

First, the total time to complete a set of concurrent simulations was measured while varying the number of concurrent simulations. Figure 9 indicates that the total time scales linearly in the number of concurrent simulations, until the point at which simulations begin to fail due to insufficient memory (points marked by “X”).

A second experiment was conducted which compares the memory footprint of a single simulation as the Java Virtual Machine (JVM) maximum heap size parameter is reduced. (There are additional elements in a JVM’s memory than the heap, but the heap limit is easy to control, with the –Xmx parameter.) As shown in Figure 10, the lowest total memory utilization achieved is 66 MB for “phase-2-init” and 111MB for “phase-2”. With a maximum heap size below 7 MB, neither simulation could complete.

Based on these values, one would expect that the current Experience Accelerator server, having 2,013 MB total RAM (181 MB used by the base system + 1,832 MB free), would support up to 27 concurrent “phase-2-init” simulations, and up to 16 concurrent “phase-2” simulations. This is consistent with the Figure 9, which showed 25 "phase-2-init" simulations completing successfully, but 30 simulations failing; and 15 "phase-2" simulations completing successfully, but 20 failing.

Extrapolating to the target of 30 concurrent simulations, one would require 2,166 MB total RAM (181 MB base + 1,980 MB free) for "phase-2-init", or 3,511 MB total RAM (181 MB base + 3,330 MB free) for "phase-2”.

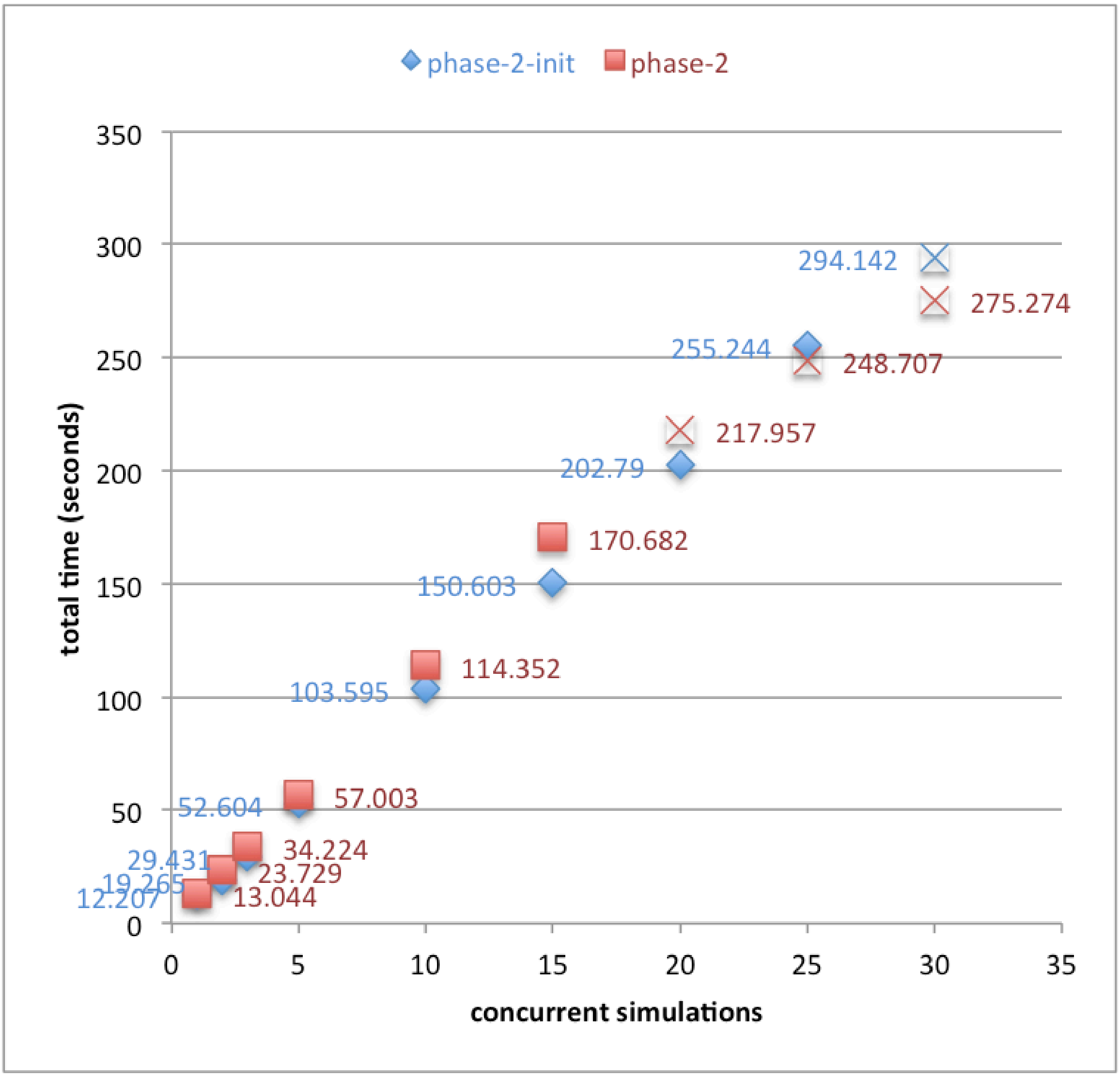


Figure 9: Total simulation time in seconds, as the number of concurrent simulations is increased. Trials marked by "X" terminated before completion, due to insufficient memory.

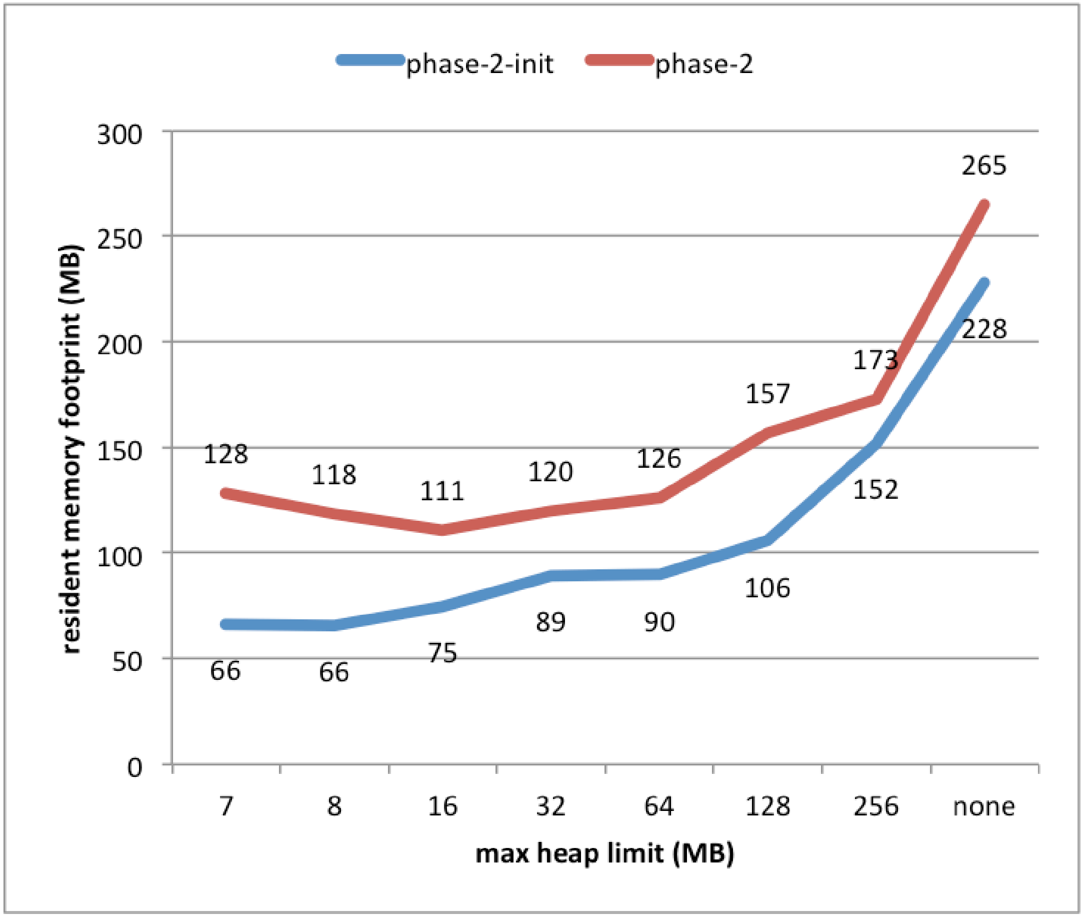


Figure 10: Resident memory footprint for a single simulation process, as the JVM maximum heap size is varied.

Figure 11 illustrates the effect of a reduced JVM heap limit on execution time. One can see that "phase-2-init" runtime performance begins to suffer when the heap limit is reduced below 32 MB, and "phase-2" runtime performance begins to suffer when the max heap limit is reduced below 64 MB. Remember from Figure 10 that with a heap limit of 64 MB, "phase 2" used approximately 126 MB of RAM. Therefore, to support 30 simultaneous simulations without an avoidable speed impact requires a minimum of 3,973 MB total (181 MB base + 30 x 126.4MB free).

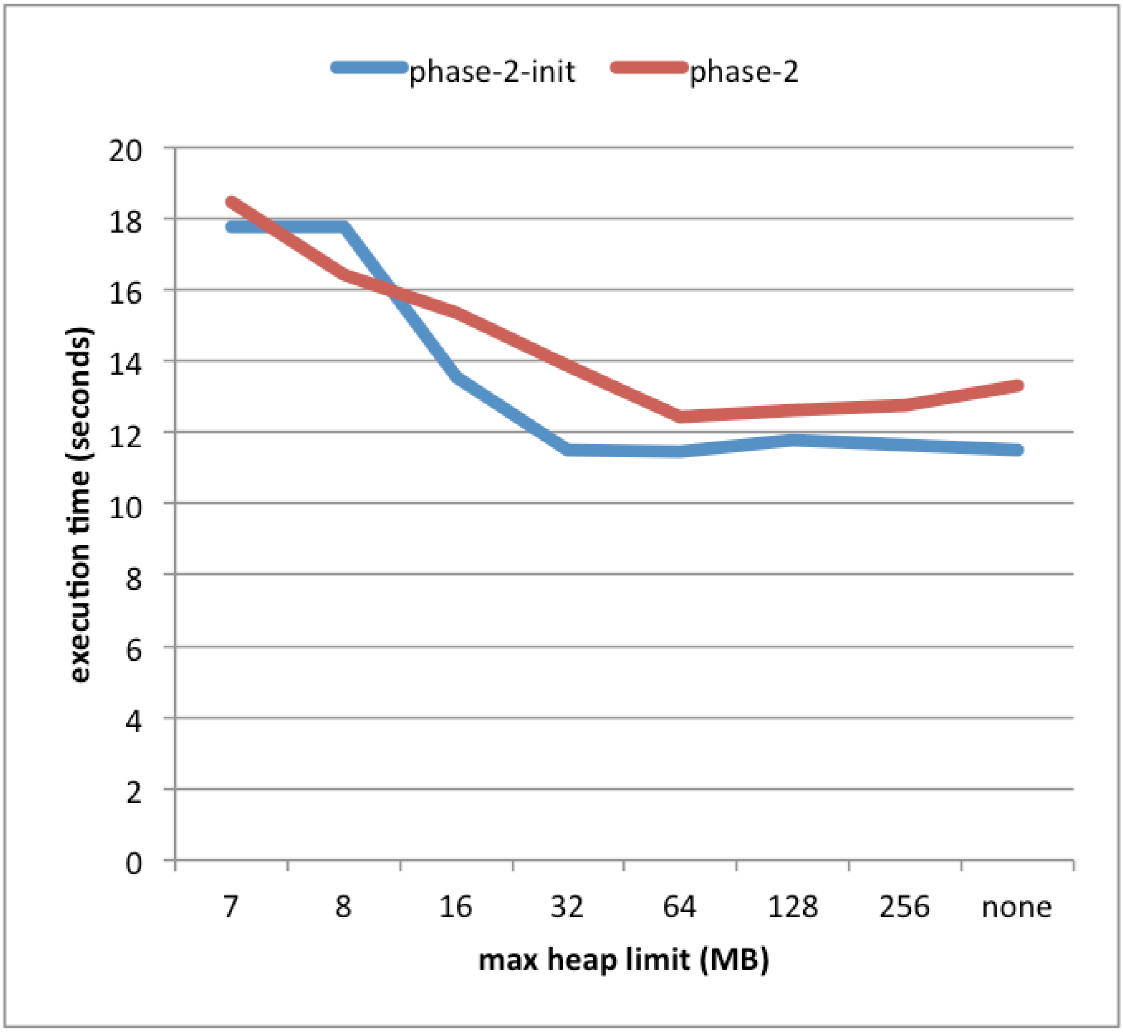


Figure 11: Execution time for a single simulation, as the JVM heap size is limited

Based on the linear trend seen in Figure 9, one would expect that even with sufficient RAM, the target of 30 simultaneous simulations would require as long as 6 minutes to complete. Figure 12 estimates the time required if more CPU cores were available: 4 cores would allow the simulations to complete in approximately 3 minutes, using 12 cores reduces the time to 1 minute, and 24 cores are needed to bring the worst case to 30 seconds. One can approximate the time for other scenarios with:

$$time = 0.4\text{ minutes} \times concurrentSimulations \div \#\, cores$$

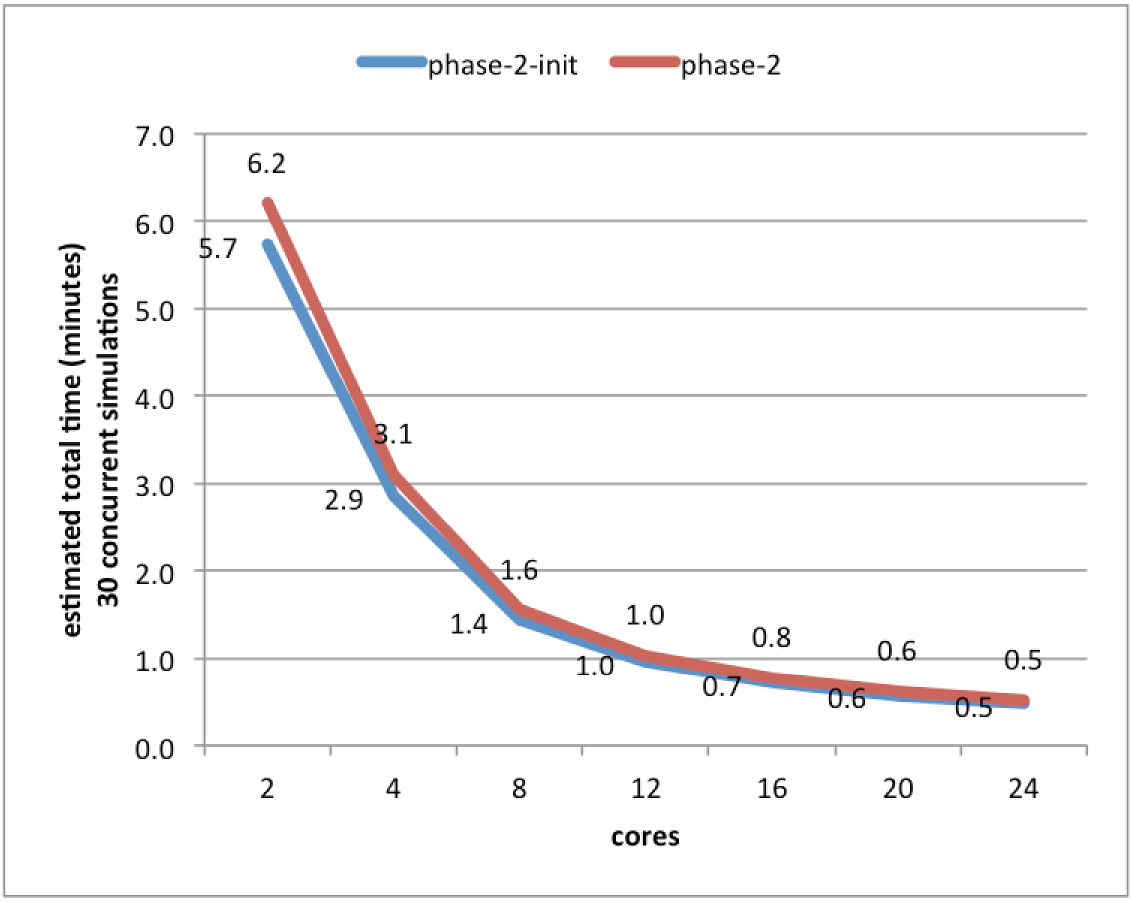


Figure 12: Estimated total execution time of 30 concurrent simulations as the number of available CPU cores is increased.

CPU utilization during the Phase-2 initialization and Phase-2 simulations is shown in Figure 13. The breakdown of CPU usage is approximately:

- 20% - startup costs (load and pre-process model XML)
- 5% - actual simulation
- 70% - rendering, encoding, and writing graph files
    - 20% reading CSV files
    - 10% drawing of the graph
    - 20% rasterizing the graph to an image buffer
    - 20% encoding the image buffer and saving as jpeg

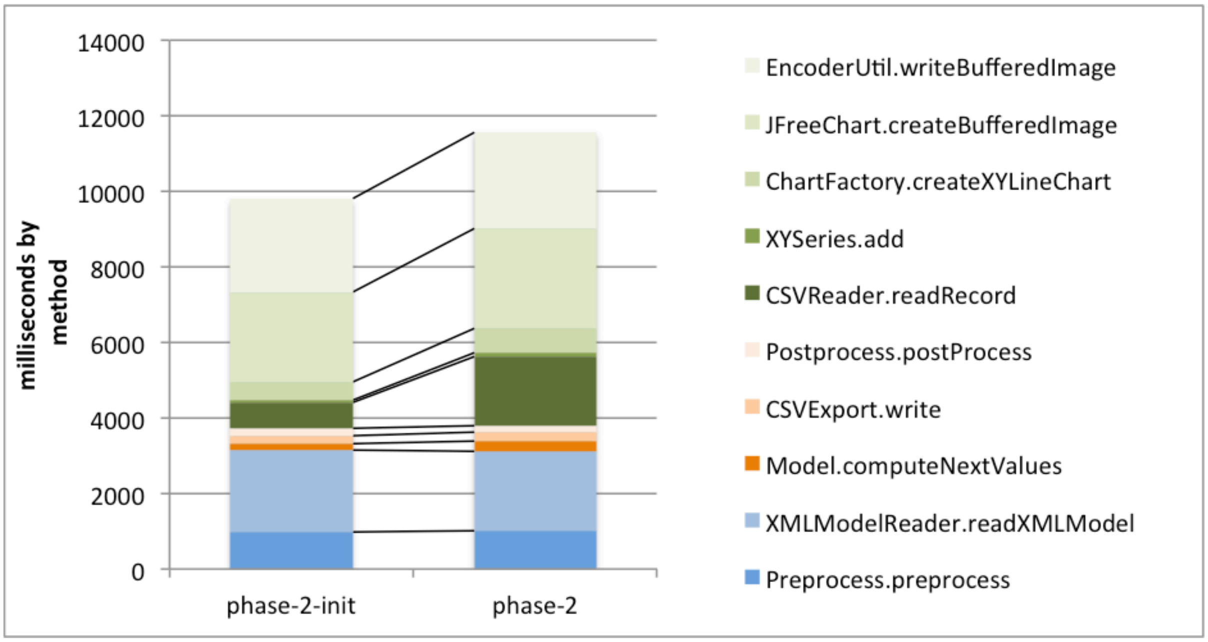


Figure 13: CPU Utilization Breakdown

2.7.2 Options for Reducing System Requirements

This section describes a number of approaches that might be taken to reduce the server resource requirements necessary to support the targeted 30 simultaneous users assuming that these users are synchronized and thus maximally stress the system.

2.7.2.1 Implement a work queue for simulation runs

Estimated level of effort: Low

Limiting the number of concurrent simulations should reduce average wait time by between 25 and 50%, depending on the number of users. In Figure 14, three users' simulation instances are started concurrently, and each takes approximately three time units to complete.

User 1 Submit	User 1 Start	User 1 Finish
User 2 Submit	User 2 Start	User 2 Finish
User 3 Submit	User 3 Start	User 3 Finish

Figure 14: Concurrent execution

In Figure 15, three users' simulation instances are started sequentially, and each takes approximately one time unit to complete. The average time to completion is two time units, instead of three.

User 1 Submit	User 1 Start	User 1 Finish	
User 2 Submit	User 2 Wait	User 2 Start	User 2 Finish
User 3 Submit	User 3 Wait		User 3 Start	User 3 Finish

Figure 15: Sequential execution

If there are underutilized cores, more instances can be run concurrently. Figure 16 illustrates six clients with a limit of two concurrent simulations.

User 1 Submit	User 1 Start	User 1 Finish	
User 2 Submit	User 2 Start	User 2 Finish	
User 3 Submit	User 3 Wait	User 3 Start	User 3 Finish
User 4 Submit	User 4 Wait	User 4 Start	User 4 Finish
User 5 Submit	User 5 Wait	User 5 Start	User 5 Finish
User 6 Submit	User 6 Wait	User 6 Start	User 6 Finish

Figure 16: Partially concurrent execution

2.7.2.2 Switch chart output format to vector-based (e.g. SVG)

Estimated level of effort: Moderate

45-50% of the measured simulator time is spent in rasterizing and encoding charts. With a vector-based chart, the GUI handles these tasks. JFreeChart only natively supports writing JPEG and PNG files, but the JFreeChart class provides a mechanism for rendering a chart onto an arbitrary AWT Graphics2D surface.

2.7.2.3 Run simulations in the EA process

Estimated level of effort: Low to Moderate

If possible, avoid spawning new processes altogether. In the case of Java processes, the HotSpot JIT compiler can optimize long-lived processes better than short-lived ones. Performing the simulations in the long-lived main process will increase the effectiveness of these JIT optimizations, as well as avoiding external JVM initialization and cleanup overhead.

(Caveat: the EA is currently insulated from any thread-safety issues that may be present in the SystemDynamics library. If any exist, they will need to be resolved before a successful integration.)

2.7.2.4 Avoid using files for inter-process communication

Estimated level of effort: Moderate

Once the simulation is integrated into the main process, there may be places where file-based communication can be avoided by directly passing the relevant data via shared memory.

2.7.2.5 Deploy multiple low-cost simulation-only servers

Estimated level of effort: Moderate

Whereas DigitalOcean leases 24-core virtual machines for $40/core, they will lease 2-core virtual machines for $10/core. Simulation and chart generation can be distributed over several machines, controlled by the master Experience Accelerator server. Because the servers would not share a common file system, Option 2.7.2.4 should be implemented first, to eliminate the dependence on the file system for inter-process communication.

2.7.2.6 Delegate chart rendering to the client application

Estimated level of effort: High

Currently, as much as 70% of the user's wait time is in producing chart image files after system dynamics computation is complete. This delay could be amortized and/or reduced by deferring chart rendering to the client application. Instead of producing chart image files, the server could send simulation result data, to be rendered by a client-side chart creation library, such as "Open Flash Chart 2". Chart formatting instructions could be generated at the server or at the client.

3 EVALUATION

3.1 INFORMAL EVALUATION

The following describes the design and results from the information evaluation of the Experience Accelerator.

3.1.1 DESIGN OF INFORMAL EVALUATION

Two forms of informal evaluation were designed in order to provide formative feedback on the EA simulation. The initial formative evaluation of the EA was a survey designed for distribution at the NDIA conference where the EA would be demonstrated. While audience members would not be able to utilize the EA themselves, they would observe as an EA team member walked them through the EA scenario, working through the experience, while explaining the goals and background of the EA as well as future plans. This evaluation focused on three areas: the audience's perception of the usefulness of the EA, their perception of the importance of different components, and their perception of how well the EA supported various features. Finally, the survey sought to recruit those interested in future testing of the EA or contributing to its development. The survey utilized a seven point Likert scale, ranging from "strongly agree" to "strongly disagree", to gather audience perceptions of the EA.

The second form of informal evaluation was a survey to be completed by the EA project's subject matter experts. This survey sought to generate much more specific feedback on various components and the interface of the EA. Likewise, it utilized a seven point Likert scale to gather the subject matter experts' perceptions of their experience utilizing the EA.

3.1.2 INFORMAL EVALUATION RESULTS

The results of the NDIA conference demonstration provided feedback from eleven audience members. The first section asked if the EA appeared as though it would be useful to their peers, students, or employees, and also asked if they liked the look and feel of the interface. Responses to the Likert questions demonstrated a strong belief that the EA would be useful and that most agreed that the look and feel was appealing. Written comments further supported that the EA was viewed as an important project for the field. Given that the EA was originally envisioned as a three dimensional environment, and was ultimately developed as a two dimensional simulated desktop environment, where the learner views emails, documents, and computer-to-computer online voice conversations, it was encouraging that the surveyed responses did not indicate that the simple visual look was a concern.

The second section of questions asked audience members to indicate the importance of components of the system to the project's success based on the description of the goals of the project. Supported competencies, the EA's architecture, the design of the experience, tools for designing experiences, supported systems dynamics, and assessment and evaluation were listed, with audience members again rating their importance on a seven point Likert scale with seven as very important and one as very trivial to the project's success. All components of the project were rated as largely important, so the audience members indicated the scope of the project was on target.

The third section asked audience members to rate their perception of how well the EA currently supported the targeted features of competencies, architecture, experience design, design tools, systems dynamics, and assessment and evaluation. Ultimately, feedback showed that overall, audience members, without having hands-on experience with the EA, perceived that the targeted features were largely supported. We also were able to identify future participants interested in evaluating or contributing to the project.

The second evaluation placed the prototype into the hands of three experienced systems engineer subject matter experts with decades of experience and asked them to evaluate the EA. While a survey was designed and provided to the subject matter experts, it was decided that the best approach to gathering their insights would be to interview them regarding their experience as they were primarily at evaluating the interface as their difficulties navigating the EA largely prevented them from evaluating other aspects of the EA in significant detail. The key result of this written and oral feedback was an expression of frustration with the usability of the learner interface. The experts struggled to navigate through the project, and this struggle overshadowed their ability to assess the level of accuracy or the effectiveness at promoting learning of the EA.

While the team had scripted guidelines for introducing users to the interface, they had not yet been developed for the interface itself, and therefore, navigating the interface proved difficult and highlighted the importance of first having these guides in place before conducting future evaluations.

3.2 FORMAL EVALUATION DEVELOPMENT

In order to evaluate the efficacy of the EA, a component of the Increment 1 EA activities focused on discovering how to evaluate the impact of the EA on learners' systems thinking competencies. In order to evaluate this impact, literature on assessing systems thinking competencies was reviewed from a variety of fields and example assessments were sought that would be well suited to a pre – post evaluation, measuring learners' systems thinking competency prior to introducing the EA and following their use of the EA. Unfortunately, nothing was found in the literature that was well suited to a quality evaluation of the EA. Therefore, a new experimental design, grounded in the literature was designed.

3.2.1 DESIGN OF EVALUATION EXPERIMENT

While the Experience Accelerator (EA) has a broader goal of accelerating the learning of critical

SE competencies through an experience-based system, systems thinking skills are a key component of the targeted learning outcomes. Critical and systems thinking is at the core of the targeted EA SE competencies and therefore one of the primary competencies to be assessed in order to evaluate the effectiveness of the EA.

Systems thinking seeks to improve decision making and complex problem solving. Typically, in order to assess learning gains in these areas, two approaches are utilized: measuring performance resulting from decisions (such as a game or simulation score), reviewing decisions and actions that were taken, or measuring learner understanding (the rules and mental operations that lead to decision making). Measuring understanding seeks to verify that improved decision-making arises.

All of these approaches are valid and can result in worthwhile evaluation. As systems- thinking skills are applied in order to understand and solve complex problems, educational research on the assessment of problem-solving skills can be helpful in designing an effective evaluation.

In order to solve an ill structured problem, students must be able to deconstruct the problem into its constituent parts (e.g., stakeholders, relationships among them, impacts of the problem on them), define the problem in their own words, determine resources to help them understand the problem, determine and pursue learning issues, and develop and test a solution. Research on the evaluation of problem-solving skills tells us that in order to evaluate problem-solving ability, we must assess students' ability to do each of these steps. The EA seeks to accelerate the learning of novice SE's and advance them more quickly to expert SE performance. Experts use heuristics to skip steps; novices typically are not capable of doing this.

A meta-analysis of problem-solving assessment literature found that 18 of 23 studies deemed high quality, used cases or simulations (Belland, French, & Ertmer, 2009). With the EA Simulation, we have the means to measure learner's performance within the experience. Learner's make decisions within the EA, the simulation determines the results of those decisions, and we are provided with outcomes that we can utilize in order to assess the effectiveness of learner's decisions.

In order to assess learners' understanding, and to determine if the EA positively impacts this (improves learning), a more thorough picture of the thinking behind learner's choices is needed. Therefore, to assess learners' understanding, we should also elicit their view of the system, the problems they faced, and the thinking behind their decision making to solve these problems. Emerging literature in systems dynamics increasingly has instead been seeking to assess learners' understanding or mental models.

Therefore, in order to assess learner performance, we can capture it (through EA results), and we can analyze the actions and decisions taken by the learner. In order to assess learner understanding, we can capture learner approaches to decision-making (through verbal protocols) and use expert choices and protocols as a baseline for "good" decision making.

Our evaluation plan therefore focused on:

- Comparing SME EA actions and results to novice SE actions and results.
- Comparing SME written (or transcribed verbal) descriptions of their decision-making process during the EA to novice SE written (or transcribed verbal) descriptions of their decision-making process during the EA in experience 1 and experience 2.

3.2.2 LEARNER IDENTIFICATION

In order to utilize learners who could provide an accurate evaluation of the EA as implemented in its intended context, Defense Acquisition University students enrolled in SYS 302 were identified as ideal for the formal evaluation. The EA will be utilized in SYS 302, replacing one case-study exercise, so the use of these students to evaluate the EA provides a perfect match between the test subjects and the target context for the EA.

The schedule utilized for the EA formal evaluation is shown below (repeated from Section 2.2.1):

Table 1: SYS 302 EA Deployment Schedule

Time	Instructors	Students
Tuesday		
1400-1430	Introduce Exercise	Team formation, role clarification/alignment
1430-1530	Mentor and Support	Individuals complete EA Phase 0 and Phase 1
1530-1630	EA PM/Mentor/Control	Team completes Phase 2A Cycle 1
1630-1700	EA PM/Mentor/Control	Team completes Phase 2A Cycle 2
Wednesday		
0800-0830	EA PM/Mentor/Control	Team completes Phase 2A Cycle 3
0830-0900	EA PM/Mentor/Control	Team completes Phase 2A Cycle 4
0900-1000	Mentor and support	Presentation Development
1000-1100	View presentations & note items for reflection material	Teams deliver presentations
1100-1130	Monitor and support	Phase 2B (Receive & discuss CDR Results)
1130-1230	Lunch (develop reflection material?)	Lunch
1230-1315	Monitor and Support	Phase 3/Phase 4/Phase 5 speed through
1315-1430	Mentoring guidance	Individuals complete Phase 7 - Reflection

Homework	Review logged results after completion	Individuals replay experience

As shown in the above schedule, test subjects utilized in the EA were placed in teams (the class was split into five teams), and completed only two full phases of the experience with the opportunity to quickly go through the remaining phases.. Each team was comprised of students playing assigned team roles. A team of 6 would be comprised of a LSE, a learner for each subsystem (4 of these) and an EVM expert. The LSE would submit the official recommendations to the EA for the team, and each team would deliver a presentation to the rest of the class, describing their choices. Subjects were then to replay the full experience individually on their own time and complete a questionnaire designed to capture their thinking. The following questionnaire was provided:

EA Student Debrief Questionnaire

In order to complete your coursework related to the Experience Accelerator (EA) systems engineering simulator, provide detailed and thoughtful answers to the following questions:

1. What strategy(ies) did your team employ during your in-class EA simulation?
2. What disagreements among your team members regarding strategies, or alternate choices during the experience were considered?
3. What ultimately led to the team's final choices during these memorable decision points?
4. What were your biggest lessons learned from the results of your team's EA session and how did these inform your approach to your at-home EA strategy?
5. What would you do differently if you were to go through the EA experience again and why?
6. Now that you have gone through the EA experience twice, once on a team and once individually, what guidelines would you provide someone else in order to be as successful as possible with the EA experience?
7. What key takeaways (lessons learned) from the EA experience do you believe you will apply to your future actual SE projects and why?

3.2.3 RESULTS

In Increment 2, the formal evaluation of the EA was not fully able to implement the designed plan for evaluation. The deployment of the EA in a pilot setting in DAU SYS 302 allowed for an ideal match with the EA's target audience. However, in order to fit within the course schedule, a number of elements of the evaluation plan had to be changed or eliminated.

Foremost amongst these were that given the considerable development required to correct existing bugs, as well as meet SYS 302 instructor needs, the DAU pilot was not completed in time to allow for a second round of evaluation from the SMEs. Furthermore, the SMEs initial evaluation of the DAU pilot was severely hampered by their difficulties with the EA's interface.

While valuable feedback was gained to improve the interface, these difficulties made it impossible to gather data from the SMEs that gave an accurate representation of their decision-making and problem-solving approaches within the EA. Therefore, data representing expert thinking was unavailable to compare with the data captured from the DAU students, who would be representing the thinking of novice system engineers.

A second significant challenge was that the questionnaire designed to capture the thinking of the target users (included in the previous section) was not completed by the students. This was a concern of the instructors who felt that after a challenging week, it was possible that the students would not be motivated to do further homework outside of class time; homework comprised of individual completion of the EA as well as a solid reflection on their thought processes while working through the EA. This proved to be the case, and so the primary data representing the thinking of the individual students, and the only data capturing their thinking after completing all phases of the experience, was unable to be gathered.

However, valuable data was still obtained in the evaluation in order to capture student thinking while completing the first two phases of the EA as well as student and instructor perspective on the EA and its efficacy. The students were observed during the two days of class that the EA was implemented, and their team presentations were also gathered. These presentations included their reflections on their lessons learned.

The targeted competencies and Aha's for the EA pilot Experience were:

- Competency 1:
    - Broad Professional category, competency #8 – Problem Solving and Recovery Approach
- Competency 2:
    - Technical Management category, competency #11 – Product Integration
- Aha 1:
    - 2.3 – Cutting corners to make short term milestones rather than focusing on end date

Based on these, the learners should have received the following lessons:

Problem solving and recovery

- Identify weight and drag problems, remediate with TPM targets and allocation changes
- Identify schedule problem, remediate with additional staff and a small schedule delay
- Identify software quality problem, remediate with increased software design review frequency

Product integration

- All sub-systems need to be done at the same time ideally for integration to begin. Adding resources to A&P and C&C sub-systems brings there schedule more in line with the other sub-system and the systems integrator, who are not having schedule issues.

- The solution could be improved by not hiring as many A&P senior staff and reducing staff for the systems integrator and ground station sub-contractor so that they meet their targets at the three-month delay, instead of the original CDR schedule. This is left as an exercise for the learner and/or instructor.
- Transferring weight allocation from A&P to C&C illustrates the relationships between these two sub-systems and how a win-win can be achieved (or at least a win and no-loss).

Cutting corners to make short term goals while ignoring long term outcomes

- Make decisions early, even though they have negative cost implications initially. This is better than facing the bigger problems later on of schedule delays that make the cost overruns worse.

The following were the actual lessons learned as presented in the five team presentations:

Team 1:
"What would we do different?

- Over correcting the Aerodynamic Drag target
  - The error was made during the 3rd and 4th cycles where another change was made to the drag target from 2.5 to 2.45, attempting to further optimize the range. During the 4th cycle no change was made to the drag target and thus we spent a lot more money on decreasing the drag and increasing the range when it was not necessary. This partially contributed to our cost overrun of 9%.
- Hire more staff early in the project to ensure there won't be a schedule slip to integration testing"

Team 2:
"Conclusions

- Based on current trends, CDR readiness NOT likely - No progress goals have been met.
- Suspect that Cycle 1 changes were not dramatic enough, in enough areas

What to do differently:

- More dramatic changes to staffing
- Leave weight targets as-is
- Identify changes more directly linked to C2 progress"

Team 3:
"Lessons Learned

- Shift staffing earlier to improve quality/test
- Reduce FTE from better performing sub-systems
- Pay closer attention to CDR entry criteria"

Team 4:

“Hindsight is 20/20

- Early software emphasis worked
- Focus on KPP early
- Maintain focus on CDR entrance criteria
- Airframe/Propulsion subsystem CPI suffered because we spent extra resources to maintain range KPP
- Our initial top 3 issues were not the main issues we encountered
- KPP was in good shape
- We got FIRED! Don’t slip CDR!”

Team 5:

“Lessons Learned

- Ramped up staff quicker
- Taken staff away from Ground Control to save money
- Raised weight limit prematurely
- Focused on drag coefficient earlier”

As can be seen, a number of the targeted lessons were clearly represented in the team presentations, an indication of the effectiveness of the EA in promoting its targeted learning outcomes.

For example, for problem solving and recovery, the importance of *small* schedule delays and the use of additional staff to remediate the schedule problem was touched on by several teams. Teams 1, 3, and 5 mentioned the need to hire staff early in their lessons learned, while Team 2 called for more dramatic staff shifts. Team 4 highlights the results of slipping the schedule too much, lamenting that they were fired for slipping CDR.

Another example can be seen with “Cutting corners to make short term goals while ignoring long term outcomes” which stresses the need to make decisions early. This was touched on repeatedly by the teams in their lessons learned. Team 1 mentions hiring more staff early, Team 3 calls for shifting staff earlier, Team 4 notes how their early emphasis on software worked, and Team 5 reflects on the need to ramp up staff more quickly.

These examples demonstrate that for these two targeted learning outcomes in particular, nearly all of the teams learned the outcomes as they very clearly highlighted them in their presentations as the lessons they had learned. Other learning objectives were also highlighted by the different teams. These lessons were learned despite the fact that the learners only fully completed the first two phases of the experience before speeding through the remaining phases in order to see their results. Furthermore, the EA was designed to be played multiple times by learners, so these results are indicative of impressive learning gains given the limited implementation of the experience.

Students also were asked to provide their perceptions of the EA – what it did well and what could be improved. The class discussed these and the comments were captured. Comments highlighted a number of key features of the EA as positives. These included the case-based format of the EA, noting its representation of real-life issues and modeling of real work interactions. Students also noted the importance of immediate feedback on the decisions and the interactive nature of the simulation – accelerating learning by simulating a project lifecycle in a short amount of time. The challenging aspect of the simulation was also highlighted as it was noted that being fired kept the learning challenging a more interesting, a possible reference to the EA's targeted "scar tissue" – emotional connection in order to promote learner transfer of learned objectives. One key positive from the feedback was that the user interface received a great deal of praise – an important change from the SME feedback.

Recommendations provided additional insights on how we can further improve the EA. For example a greater focus on performance and technical aspects as opposed to cost and schedule was highlighted. A few small bugs were also identified which will be corrected, and information which will be included in future iterations of the instructors' manual will help to further improve the efficacy of the EA.

Ultimately, the formal evaluation of the EA was hindered by the inability to compare novice learner actions and thinking to that of expert SMEs. However, while this comparison could not be made in this implementation, it certainly could be done in a future implementation. Furthermore, clear evidence of learning was gathered from the data that was gathered and learner perspectives on the efficacy of the EA were largely positive while still providing some helpful suggestions for improvement. The formal evaluation can therefore be seen as a success and indicative of the EA's efficacy in meeting its targeted learning objectives.

4 FORWARD PLAN

This section is in draft form. Rather than complete this section in isolation, a meeting with DAU sponsorship should be conducted to discuss lessons learned, future work and a risk mitigation plan which will be documented in the final version of this report.

4.1 LESSONS LEARNED

The following is a summary of the lessons learned by the RT16 team through Increment 1 and 2. The lessons are divided into the following four categories:

1. Competencies, Learning and Content
2. Complexity/Effort vs. Authenticity/Learning
3. Technology
4. R&D Processes & Tools

Lessons learned in each of these areas are described below along with an approach to mitigate negative impacts in the future. These lessons learned have impacted the nature of the future work, the processes used, and the identified risk factors moving forward.

4.1.1 Competencies, Learning and Content

LL1.1: ***Systems Thinking Evaluation*** - It is very difficult to evaluate capabilities in systems thinking. After an extensive literature search, very little was found in how to test system thinking and technical problem identification and resolution skills. Additional research will need to be done to develop a means of testing these capabilities. Our approach has been to use a Delphi approach in which subject matter experts review behavior and grade them with respect to competency levels.

LL1.2: ***Learning and Concept Capability Evaluation*** – It is difficult to determine if the learner has actually learned and can apply the identified concepts. Even if the learner goes through the Experience more than once and shows improvement, it is not clear whether they have just learned how to best this particular experience or if they have learned the concept and how to apply it in future situations. It would be desired to develop and exam in a different medium which could be used for pre- and post-testing to assess the results.

LL1.3: ***Use of Subject Matter Experts*** - The project has relied on subject matter experts to help authenticate and validate the experience. The subject matter experts have extensive experience in aircraft systems design and development. On the other hand, the experience is intended for use in a current "state-of-the-art" course in systems engineering at Defense Acquisition University. The intent is this course is to teach the most up-to-date methods for systems engineering in DoD acquisition. There were several instances where input from SMEs conflicted with methods taught at DAU, particularly in the presentation of TPMs for the redesign of the legacy airframe used for the UAV. In the future, it is advisable that experience be developed with simultaneous SME and DAU/instructor feedback.

LL1.4: ***User Interface*** - Learners expect a smooth, relatively rapidly updated user interface that is similar to their other tools.

LL1.5: ***Importance of Understanding Lags*** - Learners need to understand the lag between changing something and the actual evidence the change had the appropriate result (discuss the Dietrich Dorner "Logic of Failure" results).

LL1.6: ***Importance of Background Information*** – Learners require more background information to make decisions than was originally anticipated.

LL1.6: ***Technical vs. Programmatic Issues*** – The current experience focuses more on the programmatic than on the technical issues in a program. It is an open question on how this can be addressed while still working within the restricted amount of time that is available for the experience while being flexible with the level of domain knowledge possessed by the learner.

LL1.7: ***Role Adaptation*** – It is desired to have the ability to adapt scenarios for different SE roles (LSE on gov't side very different from LSE on developer side).

LL1.8: ***Impact of Non-Player Characters (NPCs) Roles*** – NPCs are valuable, but their role title may impact usefulness (learners did not want to communicate with the senior Program Manager). However, this real world behavior with respect to NPCs provides the opportunity for additional learning.

LL1.9: ***Decision Making Tools*** – Learners desire "what-if" tools to help with decision making. Is this helpful or is failure resulting from trial and error a better educational method?

4.1.2 Technology

LL2.1: ***Client Graphic Technology Migration*** - While Flash is currently has the most productive environments in which to develop graphical content and is free on the client, the development licensing can be expensive (approximately $200 for individual educational licenses, and $700 for individual commercial licenses) and Flash does not work on iPads which represent a major client technology base. Once the open competing technology, HTML5, has development environments which provide the same level of productivity of Flash, it would be advantageous to migrate to the new technology.

LL2.2: ***Client/Server Interface Reliability*** - We had a number of problems getting the EA client/server interface reliably in the presence of an unreliable network. While the system now works, getting it to this state required a significant amount of debug and redesign work. In Increment 2, we will need to review the design and update it and the implementation to provide a cleaner, more supportable system.

LL2.3: ***UI Look and Feel*** - Creating a professional look and feel for a virtual desktop continues to be a non-trivial undertaking. We received feedback from the subject matter experts on the difficulties that they faced in using the browser and virtual desktop. We have added a text based support online, but we may need to create an overall experience training video as well. We have made and will continue to make changes to address this.

4.1.3 Complexity/Effort vs. Authenticity/Learning

LL3.1: ***Challenges & Landmines*** - There are an almost infinite number of ways in which a program can fail; combinatorial explosion is a major challenge. This is not so much of a challenge for the simulator, but this continues to be a major issue for the creation of artifacts and dialog which can support these to allow the Learner to make sense of the situation. While we created a catalog of a large number of frequently encountered Challenges and Landmines, for the prototype we implemented just a few of the most likely ones in the areas of aviation hardware and software. During this past year we have added challenges in the area of budget management.

LL3.2: ***KPMs*** - KPMs drive the amount of information that needs to be simulated, the amount of artifacts for background information, dialog and Learner recommendations. During Increment 1 we have added cost to the existing KPMs of schedule, quality and capabilities (range) which has required the development of a plausible cost model and supporting artifacts and dialog. While we have implemented a relatively simple EVM system, it is clear that developing one similar to what would be found in a DoD program office would be very challenging. We will have to see if what we developed has sufficient authenticity to provide the desired learning.

LL3.3: ***Feedback to/from Learner*** – In Increment 2 we increased the amount of feedback to the learner in the form of a confirmation of the actions which were taken as a result of the recommendations that the learner made throughout the experience, and in the feedback that the learner received at the end of the experience. While the recommendation form of feedback was fairly straightforward in its implementation as it represented objective fact, the reflection feedback was more complicated in that it was both subjective in nature and the multiple actions that the learner has taken are inter-related. The process that was used involved noted the decisions that were deemed appropriate, inappropriate and neutral and feedback on each decision was made independently. Understanding the interactions between these is far more complex. One possibility for future work is to record how the learner's behave and determine if there are specific patterns of behavior which can be identified and feedback given to the learner which is most appropriate to the pattern of behavior.

LL3.4: Balance of Complexity and Authenticity in Defense Acquisition - Defense acquisition is a very complex enterprise, with many processes, actors and organizations. Selecting a subset of these to represent in the Experience Accelerator involves numerous design trade-offs to address the interests of (i) the learner community, that wants a realistic but not overwhelming experience, (ii) the education community, that wants realism in support of learning objectives, (iii) the acquisition community, that wants its various aspects represented faithfully, and (iv) the developer community, that wants to provide a useful product while managing complexity, schedule and cost. One of the biggest challenges was to create an authentic, learning experience while managing complexity and the amount of content that needed to be created. During this past year we have continued to learn in these areas.

LL3-5: Extending the Learner's Role to Improve Learning - One of the recommendations that the learner can make is to change the labor resources assigned to sub-system development. Since the learner is the government's lead systems engineer, it was pointed out by DAU faculty that this recommendation choice is not realistic. The government LSE cannot make staffing recommendations for contractors, although he or she can have a discussion with the prime contractor LSE about current deficiencies and alternatives (and costs) to remedy them, including the contractor deciding to change staffing. The decision was made to keep the LSE's ability to recommend staffing changes as a surrogate for the more complex discussions that he or she might have with the prime's LSE. This might be revisited in the future with richer NPC interactions.

LL3-6: User Interface for Data Navigation - The number of simulation output charts tends to increase with the number of stakeholders involved in review and usage, since each stakeholder has potentially a different perspective on important metrics to track. This makes navigating the chart set difficult for a user. The interface for such navigation needs to be designed and implemented.

4.1.4 R&D Processes & Tools

At the outset of this project, given the exploratory nature of the research, open communication was established between all of the team members through the use of a Wiki site. However, it was discovered that the overhead in learning how to use and navigate through the site outweighed its advantages so we migrated to the use of a simple DropBox technology for documents and development code, and Webex for group meetings which were held once a week. As the team more than doubled in size and became more specialized, this mode of communication became a bit cumbersome. We eventually migrated to a weekly meeting with the SMEs for content, a weekly team meeting and technology meetings on an as needed basis. While the intention was for the team to develop technology and content using an agile software development process on a single server site at Purdue, it was more difficult than expected to set up an iterative development environment and workflow. During the Increment 1, we moved the development to Stevens and used a set of technologies for bug tracking and version control that are standard in the open source software world. The following are some of the lessons that were learned in Increment 1 and 2 after moving to this new environment.

LL4.1: ***Change Management*** - In Increment 1 we started using a formalized ticketing system and process. Due to the instability of the design, we quickly created a huge number of tickets for the developers. Due to this large backlog, we slowly moved away from this system, and started contacting the developers directly on our needs. In retrospect, this was a mistake as it created challenges in tracking the work in queue, in process and completed. Now that the design and implementation has stabilized, we intend to move back to the more formal approach that should prepare for open source distribution and should help us with our challenges in configuration management (see LL4.2).

LL4.2: ***Configuration Management*** - Due to the large number of design files, artifacts, simulation parameters and the like, configuration management has become a great challenge. One of the issues is that the Experience Design document is overly generalized and it is sometimes difficult to know exactly what has been implemented. Another issue is that we do not have a centralized place that is a source for all of the files. The issue goes beyond the software build and reflects all of the artifacts and code. To address this issue we will be creating a single design document that will be used with a work tracking tool to provide configuration management for the program. Reconciliation and updates in these two sources of data will be done on a weekly basis to ensure that control is maintained over the design.

LL4.3: ***Verification and Validation*** - The system that we are developing has become very complex making it quite difficult to test. In addition, resources are quite limited such that the testing that we do is ad hoc and carried out by the person who is implementing the code. One possible solution to this problem is to develop an automated tool for verification. There are simulation packages available that have the ability to automate tests that provide the ability to run a suite of simulations and then check all of the results in parallel. In addition, we should create a pool of people who are outside the implementation team who can test the overall Experience Accelerator system.

LL4.4: ***Configuration Dependency Testing*** - There are a number of implementation issues that are configuration dependent, yet may not be discovered through our normal testing procedures which are limited in their configuration variations. Addressing this will require automated test beds and the prerequisite software and hardware support.

LL4.5: ***Content Creation*** - It is quite difficult to create content, particularly dialog, without actually going through the experience to see what it is like. Unfortunately, this was not possible during much of Increment 1 due to instabilities in the code base and the recent development of the EVM features. In addition, as a content creator you need the ability to see the effects of the experience from the beginning to the end. It would be quite useful to have a developer mode in which one could quickly see the projected results of an entire Experience based on some scripted learner behaviors. These behaviors could be based on recorded behaviors of actual learners. This information could also be useful in the identification of classes of behavior types, how to identify them, and how to improve their performance.

LL4.6: ***Architectural Conformance*** - There was a lack of conformance between the implementation, heavy client, and the architecture, thin client, that was developed in the base line year. As a result, there was a good deal of rework that had to be completed by the Stevens development team. It was clear that we needed someone with a computer science and game design background to manage the development team. This action was taken and the issues have been remedied.

LL4.7: ***Academic Software Development*** - Software is difficult in an academic setting as the workforce is largely composed of students who quickly come and go. In addition, since the development team is so small, consisting of 2-3 member, transitions are particularly difficult as the loss of a single member is a large fraction of the team, there is little overlap with other members, and it is not easy to quickly recruit and bring up to speed new members. It is clear that there needs to be at least one member of the team who understands the entire design and is there long term to provide continuity to the team. This has been accomplished on the technology, simulation and experience design teams through continuity of technically capable faculty researchers. In addition, longer term support is also in place with a graduate student in technology development.

LL4.8: ***Automation of Repetitive Tasks*** - There are a number of repetitive tasks that consume a great deal of our developers' time and effort. One example is the conversion of documents from one format to another which then need to be placed in the appropriate design file. There is a strong need for the development of tools to automate these tedious processes. There is a need for tools to automate tedious processes. A conceptual design has been created for this tool.

LL4.9: ***System Interfaces and Partitioning*** - Another lesson learned in the baseline year and repeated in Increment 1 is the importance of partitioning in the system and creating interfaces such that artifacts and dialog can be created without the involvement of developers. This was increasingly successful in Increment 1 as tools were created to allow the simulation team to create graphs and charts that could be automatically incorporated into the system

LL4.10: Simulation Tuning - For the DAU pilot, the difficulty of the experience was called back somewhat so that the learner could succeed with a 20% cost overrun and three month schedule slip in Phase 2 if the correct decisions are made. While the difficulty level can be controlled by manually tuning the simulator, this approach will not scale up to handle numerous difficulty levels targeting different experience aspects. A more automated approach is needed.

4.2 FUTURE WORK

The plan is to preserve most of the EA team going in moving the Experience Accelerator to DAU deployment, open source distribution and community development.

Follow-on work has been defined for Increment 3 that is focused on the following:

- EA System Capabilities
    - Completion and stabilization of multi-learner mode
    - Provide means of informing learner of impact of recommendations
    - Ensure that dialog is synchronized with recommendations
    - Improve learner interface with status charts to eliminate need to page through entire set
- Tools
    - Create set of tools that allow the DAU to customize and create new Experiences
- Deployment Deliverables
    - Define explicit EA deliverables to support DAU deployment
- Hosting Requirements
    - Specify technical details of hosting requirements

This work is targeted to support the following DAU Pilot schedule:

- 01/xx/2014 Instructor Pilot
- 02/04-05/2014 Student Pilot
- 03/19/2014 Instructor Pilot
- 04/08-09/2014 Student Pilot
- 06/26-27/2014 Instructor Pilot (SYS-30X full course)
- 08/12-13/2014 Student Pilot
- 09/18-19/2014 Student Pilot

For more detail on the follow-on work plan, see *Developing Systems Engineering Experience Accelerator (SEEA) Prototype and Roadmap, Increment 3 Technical and Management Work Plan.*

In addition to potential Increment 3 funding, significant outreach efforts have been made during Increment 2 to find additional sources of funding for the current research team, and opportunities for joint research and external research and development that is necessary to support and sustain an open source community for the Experience Accelerator technology and content. This research falls into two major categories, the development of new capabilities that Increment 3 supports in further refinement of the current experience and multi-learner technology and new experiences. Some of the opportunities that are being explored are noted below:

- Extended Capabilities
    — SERC: Content Creation Tools funding
- New Experiences
    — DAU: Logistics Experience, proposal submitted
    — ONR: Team experience, white paper submitted
    — NSF: Learning in Formal and Informal Settings, 1/14/13 submission
    — NRO: Spacecraft experience, will pilot SEEA
    — MITRE: Team experience, discussions
    — BAE: early exploration
    — Sponsored doctoral research: 2-3 Stevens students

In Increment 3, efforts will be increased to expand the research efforts both internally and externally so that significant development is taking place at multiple research sites to enable our long-term objective of a sustainable open source Experience Accelerator community.

4.3 Risk Management

The following Risk Management plan for Increment 2 needs to be updated based on Sponsor feedback and discussion.

4.3.1 RISK 1: PROJECT MANAGEMENT

Risk: Inability to support known and evolving customer/user feedback with current staff, budget and timeframe.

Mitigation: No significant new EA features are targeted for Increment 2, rather new capabilities will be restricted to those that address the feedback that we receive from learner evaluations of the Experience Accelerator. The work will be targeted at improving the current system to make it ready for Piloting.

4.3.2 RISK 2: CONFIGURATION MANAGEMENT

Risk: Inability to successfully manage the large number of files, configuration variables, present in the Experience Accelerator. (See LL4.2)

Mitigation: A more formalized approach will be used to provide assurance that the implemented configuration is in compliance with the desired specifications. This will be accomplished by creating a single design document that will be used with a work tracking tool to provide configuration management for the program. Reconciliation and updates in these two sources of data will be done on a weekly basis to ensure that control is maintained over the design.

4.3.3 RISK 3: TECHNOLOGY DEVELOPMENT

Risk: Inability to tradeoff long-term architecture and technology objectives (leading to successful open source support) vs. short-term prototype goals. One long term issue is the reliance on Flash which is a proprietary standard not being supported in some Apple client devices (e.g., iPad and iPhone). The likely standard to be supported in the future is HTML5. The time to make the transition will be dependent on when the toolkits are available to make it a productive environment and when it supported on browsers in use (the DoD used some old IE versions which do not support it).

Mitigation: Supporting this migration will require additional funding, with the timing of the migration likely to be post Increment 2. In the meantime, we will keep track of the evolving standards and attempt to reduce the impact of the eventual migration.

4.3.4 RISK 4: CONTENT DEVELOPMENT

Risk: Inability to produce a prototype that provides a compelling experience, supports the desired learning and is seen to be authentic.

Mitigation: Develop and review a design experience document which is used to guide the development process. This experience document will be improved to ensure that it contains the specific information necessary to facilitate configuration management. Unfortunately, due to the instability of the implementation in Increment 1, it was difficult to iteratively develop dialogue and feedback. However, the Experience Accelerator is now sufficiently robust that this iterative approach can be taken. Additional tools will be explored that could improve this situation by providing the ability to quickly see the ramifications of specific learner behaviors.

4.3.5 Risk 5: Evaluation

Risk: Inconclusive results due to threats to validity of Experimental design (inability to generalize results), limited availability of suitable subjects and insufficient literature to support development of evaluation instruments. The critical challenge is to determine how to measure success in systems thinking and problem identification and resolution.

Mitigation: Additional work will be done to synthesize the published results in the literature. Explore development of new research instrumentation by synthesizing relevant literature should no suitable instrumentation be found in the literature. Create the capability to collect and analyze learner behavior traces, and compare pre- and post-experience traces of learners versus those of acknowledged experts. Possibly utilize Delphi sessions with SMEs will be explored as a means to develop a set of tests that can be used for pre- and post-Experience evaluation in these areas.

References

Bagg, T. C., Granata, R. L., Brumfield, M. D., Casey, C. A., & Jamison, D. E. (2003). "Systems engineering education development (SEED) case study." In *Proceedings of the 13th annual international symposium, INCOSE 2003, Crystal City, Virginia, USA, June 29 - July 3, 2003.*

Bodner, D., Wade, J., Squires, A., Reilly, R., Dominick, P., Kamberov, G., Watson, W. (2012), "Simulation-Based Decision Support for Systems Engineering Experience Acceleration", IEEE Systems Conference, Vancouver, BC, Canada, March 19-23.

Bodner, D., Wade, J. (2013) "Multi-Criteria Simulation of Program Outcomes," Proceedings of the 2013 IEEE International Systems Conference, 215-221, Orlando, FL, April 15-18, 2013.

Bodner, D., Wade, J., Watson, W., Kamberov, G (2013) "Designing an Experiential Learning Environment for Logistics and Systems Engineering," Proceedings of the 2013 Conference on Systems Engineering Research (CSER), Atlanta, GA, March 20-22, 2013.

Charette, R. (2008). "What's wrong with weapons acquisitions?" *IEEE Spectrum*, November.

Defense Acquisition Program Support (DAPS) Methodology. (2009). *Defense Acquisition Program Support (DAPS) Methodology* (2nd ed. p. 442). Washington, DC: Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. Retrieved from http://www.acq.osd.mil/se/docs/DAPS_V2.0_Methodology.pdf

Dubey, R. (2006) "Study and Analysis of Best Practices for the Development of Systems Engineers at a Multi-National Organization." *Thesis*, MIT, Boston, MA, USA.

Gavito, V., Verma, D., Dominick, P., Pennotti, M., Giffin, R., Barrese, T., et al. (2010, December 13). “Technical leadership development program: Final technical report SERC-2010-TR-013.” Stevens Institute of Technology, Hoboken, NJ.

Kolb, D. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall, Englewood Cliffs, NJ.

Madachy, R. (2008). *Software Process Dynamics*. Washington, DC: Wiley-IEEE Press.

NDIA SE Division (2010, July). “Top systems engineering issues in department of defense and defense industry (Final 9a-7/15/10).”

Ryschkewitsch, M., Schaible, D., & Larson, W. (2009, January 20). “The art and science of systems engineering: Long version.” *NASA.* Retrieved March 1, 2009, from http://www.nasa.gov/pdf/311198main_Art_and_Sci_of_SE_LONG_1_20_09.pdf

Squires, A. (2007). “Qualifying Exam: Systems Thinking and K12 Education.” *Stevens Institute of Technology,* Hoboken, NJ.

Squires, A., Wade, J., Watson, W., Bodner, D., Reilly, R., Dominick, P. (2012), “Year One of the Systems Engineering Experience Accelerator”, Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER), Rolla, Missouri, March 19-22, 2012.

Squires, A., Wade, J., Watson, B., Bodner, D., Okutsu, M., Ingold, D., Reilly, R., Dominick, P., Gelosh, D. (2011), “Investigating an Innovative Approach for Developing Systems Engineering Curriculum: The Systems Engineering Experience Accelerator”, Proceedings of the 2011 American Society for Engineering Education (ASEE) Annual Conference and Exposition, Vancouver, BC, Canada, June 26-29, 2011.

Squires, A., Wade, J., Dominick, P., Gelosh, D. (2011) “Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers”, Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER), Redondo Beach, CA, April 14-16, 2011.

Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston: McGraw-Hill.

Wade, J., Kamberov, G., Bodner, D., Squires, A. (2012) “The Architecture of the Systems Engineering Experience Accelerator”, International Council on Systems Engineering (INCOSE) 2012 International Symposium/European Conference on Systems Engineering (EUSEC), Rome, Italy, July 9-12.